PrintShop Mail Connect

User Guide

Version 2022.1.5

Objectif Lune

PrintShop Mail Connect

OL® Software

User Guide
Version 2022.1.5
Last Revision: 2023-05-22

Upland Objectif Lune
2030 Pie-IX, Suite 500
Montréal, QC, Canada, H1V 2C8

+1 (514) 875-5863
www.objectiflune.com

All trademarks displayed are the property of their respective owners.

© Upland Objectif Lune. 1994-2023. All rights reserved. No part of this documentation may be reproduced, transmitted or distributed outside of Upland OL by any means whatsoever without the express written permission of Upland OL. Upland OL disclaims responsibility for any errors and omissions in this documentation and accepts no responsibility for damages arising from such inconsistencies or their further consequences of any kind. Upland OL reserves the right to alter the information contained in this documentation without notice.

Table of Contents

Table of Contents **4**

Welcome to PrintShop Mail Connect 2022.1 ..... **12**

Setup And Configuration **13**

System and Hardware Considerations 13

Antivirus Exclusions 13

Database Considerations 15

Environment considerations 18

Language and Encoding Considerations 20

Network Considerations 21

Performance Considerations 23

System requirements 25

Installation and Activation 28

Where to obtain the installers 28

Installation prerequisites 29

User accounts and security 29

Installing PrintShop Mail Connect on Machines without Internet Access 30

Installation Wizard 32

Running Connect installer in Silent Mode 38

Activating a License 45

Migrating to a new workstation 48

Upgrading 54

Upgrading from Connect 2021.2 or earlier with Remove User Data CHECKED: 57

Upgrading from Connect 2021.2 or earlier with Remove User Data UNCHECKED: 58

Uninstalling Connect 2022.1 or above with Remove User Data CHECKED: 58

Uninstalling Connect 2022.1 or above with Remove User Data UNCHECKED: 59

Preferences 60

Clean-up Service preferences 60

Database Connection preferences 64

Editing preferences 67

Email preferences 71

Emmet preferences 73

Engines preferences 76

Hardware for Digital Signing preferences 76

Language preferences ... 77
Logging preferences ... 77
Parallel Processing preferences ... 80
Print preferences ... 80
Sample Projects preferences ... 81
Save preferences ... 82
Scripting preferences ... 83
Engines preferences ... 84
Parallel Processing preferences ... 88
Known Issues ... 90
Issue with CRLF in Go Sub runtime parameters ... 90
Installer issues ... 90
Dynamic tables causing issues ... 90
DataMapper: Automatic Date/Time does not work with certain negative UTC time zone offsets ... 91
CSS inlining colour values now converted to RGB ... 91
Issues running Connect on Hyper-V 9.0 ... 91
Minor font changes ... 92
Uninstalling ... 100
General information ... 102
Connect: a peek under the hood ... 102
The Connect database ... 103
The File Store ... 103
The engines ... 104
Log files ... 104
Location ... 104
Name ... 105
Format ... 105
Connect file types ... 105
The Designer ... 107
Designer basics ... 107
Features ... 108
Templates ... 109
Contexts ... 118
Sections ... 119
Print ... 124
Creating a Print template with a Wizard ... 126

Print context 132
Print sections 137
Pages 147
Master Pages 156
Media 159
Email 166
Designing an Email template 167
Creating an Email template with a Wizard 171
Email context 174
Email templates 177
Email header settings 180
Email attachments 187
Content elements 191
Element types 191
Editing HTML 193
Attributes 193
Inserting an element 194
Selecting an element 196
Deleting an element 196
Styling and formatting an element 197
Barcode 197
Boxes 256
Business graphics 260
Date 267
Hyperlink and mailto link 268
Images 271
Table 278
Text and special characters 283
Snippets 285
Adding a snippet to the Resources 285
Creating a snippet 287
Adding a snippet to a section 287
Editing a snippet 287
Renaming a snippet 287
Translating a snippet 288
HTML snippets 288
JSON snippets 290

Handlebars templates ... 291
Styling and formatting ... 294
Local formatting versus style sheets ... 295
Layout properties ... 295
Styling templates with CSS files ... 296
Styling text and paragraphs ... 308
How to position elements ... 312
Rotating elements ... 315
Styling a table ... 316
Styling an image ... 320
Background color and/or image ... 323
Border ... 325
Colors ... 328
Fonts ... 332
Locale ... 336
Spacing ... 337
Personalizing content ... 338
Variable data ... 339
Conditional content ... 339
Dynamic images and Print section backgrounds ... 339
Snippets ... 340
Scripts ... 340
Loading data ... 341
Variable Data ... 351
Formatting variable data ... 357
Showing content conditionally ... 358
Conditional Print sections ... 363
Dynamic images ... 366
Dynamic Print section backgrounds ... 368
Preferences ... 370
Clean-up Service preferences ... 370
Database Connection preferences ... 374
Editing preferences ... 377
Email preferences ... 380
Emmet preferences ... 382
Engines preferences ... 385
Hardware for Digital Signing preferences ... 385

Language preferences 386
Logging preferences 387
Parallel Processing preferences 389
Print preferences 390
Sample Projects preferences 391
Save preferences 391
Scripting preferences 392
Writing your own scripts 393
Script types 394
Creating a new Standard Script 395
Writing a script 397
Setting the scope of a script 399
Managing scripts 400
Testing scripts 403
Optimizing scripts 408
The script flow: when scripts run 413
Selectors in Connect 414
Loading a snippet via a script 420
Loading content using a server's API 423
Using scripts in Dynamic Tables 426
Control Scripts 429
Post Pagination Scripts 445
Handlebars in OL Connect 452
Translating templates 467
Translating a template 468
Tagging elements for translation 469
Pluralization 472
Exporting and importing translation files 474
Designer User Interface 476
Dialogs 477
Keyboard shortcuts 561
Menus 568
Panes 577
Toolbars 598
Welcome Screen 603
Print options 604
Job Creation Presets Wizard 684

Output Creation Presets Wizard 700
Rule name 798
Group 798
Rule 799
Expand button 800
Designer Script API 809
Standard Script API 810
Functions and fields 837
Example 837
Functions and fields 838
html() **839**
html() 839
html(value) 839
margins **840**
Fields 840
Fields 841
front, back **842**
Control Script API 931
Post Pagination Script API 967
Generating output **992**
Print output 992
Email output 993
Generating Print output 993
Print settings in a template 995
Aborting content creation 996
Print using standard print output settings 996
Print Presets 998
Print using Advanced Printer Wizard 1005
Adding print output Models to the Print Wizard 1005
Splitting printing into more than one file 1007
Print output variables 1008
Generating Email output 1017
Before generating Email output 1018
Testing Email output 1020
Aborting content creation 1021
Using an ESP with PrintShop Mail Connect 1021

Optimizing a template 1026
Scripts 1027
Images 1027
Print Manager 1029
Print Manager Introduction Video 1029
Print Manager usage 1029
Print Manager Interface 1030
Menu options 1030
Printers pane 1031
Printer Queue pane 1038
Job Queue pane 1047
Messages pane 1049
Job History pane 1050
Printer Capabilities pane 1052
Printer Status pane 1052
Print Manager Preferences 1054
PrintShop Mail Connect Release Notes 1064
OL PrintShop Mail Connect Release Notes 2022.1.5 1064
License Update Required for Upgrade to OL Connect 2022.x 1064
Backup before Upgrading 1065
Overview 1065
OL Connect 2022.1.5 Improvements 1067
OL Connect 2022.1.4 Improvements 1067
OL Connect 2022.1.2 Improvements 1067
OL Connect 2022.1 Improvements 1069
OL Connect 2022.1 Designer Improvements 1072
OL Connect 2022.1 Output Improvements 1076
OL Connect 2022.1 Print Manager Improvements 1078
OL Connect 2022.1 Improvements 1079
Known Issues 1079
Previous Releases 1079
OL PrintShop Mail Connect Release Notes 2021.2.1 1079
OL PrintShop Mail Connect Release Notes 2021.1 1089
OL PrintShop Mail ConnectRelease Notes 2020.2.1 1096
OL PrintShop Mail Connect Release Notes 2020.1 1109
OL PrintShop Mail Connect Release Notes 2019.2 1117

OL PrintShop Mail Connect Release Notes 2019.1 1135
PrintShop Mail Connect Release Notes 2018.2.1 1151
PrintShop Mail Connect Release Notes 2018.1.6 1178
PrintShop Mail Connect Release Notes 1.8 1201
PrintShop Mail Connect Release Notes 1.7.1 1222
PrintShop Mail Connect Release Notes 1.6.1 1243
PrintShop Mail Connect Release Notes 1.5 1251
PrintShop Mail Connect Release Notes 1.4.2 1258
Connect 1.4.2 Enhancements and Fixes 1259
Connect 1.4.1 New Features and Enhancements 1259
Connect 1.4.1 Designer Enhancements and Fixes 1261
Connect 1.4.1 Output Enhancements and Fixes 1262
Known Issues 1263
Knowledge Base 1266
Legal Notices and Acknowledgements 1267
Copyright Information 1275

Welcome to PrintShop Mail Connect 2022.1

Note

Since we are always looking for new ways to make your life easier, we welcome your questions and comments about our products and documentation. Use the feedback tool at the bottom of the page or shoot us an email at doc@ca.objectiflune.com.

PrintShop Mail Connect is a tool designed to optimize customer communications. It is designed to improve the creation, distribution, interaction and maintenance of your communications.

The PrintShop Mail Connect **Designer** is designed to create output for print and email within a single template and from many data types. Output presets applied outside the design phase make templates printing device independent.

The **Designer** has an easy-to-use interface that makes it possible for almost anyone to create multi-channel output. More advanced users may use native HTML, CSS and JavaScript.

This online documentation covers **PrintShop Mail Connect** version 2022.1.

Setup And Configuration

This chapter describes the PrintShop Mail Connect installation and the different considerations that are important in regards to the installation and use of PrintShop Mail Connect.

- "System and Hardware Considerations" below
- "Installation and Activation" on page 28
- "Known Issues" on page 90
- "Engines preferences" on page 84

System and Hardware Considerations

There are a variety of considerations to be aware of. These are documented in the following pages:

- "System requirements" on page 25
- "Database Considerations" on page 15
- "Environment considerations" on page 18
- "Known Issues" on page 90
- "Language and Encoding Considerations" on page 20
- "Antivirus Exclusions" below
- "Performance Considerations" on page 23

Antivirus Exclusions

The information on this page is designed to assist IT managers and IT professionals decide what anti-virus strategy to follow with consideration to PrintShop Mail and their internal requirements and needs. This page describes the mode of operation and the files and folders used by PrintShop Mail as well as the files, folders and executables that are recommended to be ignored for best possible performance and to avoid issues caused by antivirus file locks.

IT managers and IT professionals then may decide the anti-virus strategy to follow for their internal requirements and needs depending on the statements outlined herein.

Directories and folders

All Connect applications are installed under an arbitrarily selectable main folder. If the default installation folder options were used, this installation folder would be `%PROGRAMFILES%\Objectif Lune\OL Connect`.

The installation folder will hold all the executable files and other files and folders required for the operation of the whole product suite. All these files and folders remain static after installation. It depends upon the company virus protection strategy, if such files and folders will be monitored or not.

We do, however, recommend that the following file or folders be ***excluded*** from antivirus protection.

Working folders

Working folders for Connect are created and used on a per-user-basis under the respective user's profile folder, accessible on Windows with the standardized system variable `%USERPROFILE%` in the subfolder "Connect". Working folders are:

- `%USERPROFILE%\Connect\filestore`: This folder will hold non-intermediate files for the operation of Connect. Files in this folder will be used frequently, but not with a high frequency. Supervising this folder with a virus protection system should not have too much of an impact on the speed of the whole Connect suite.
- `%USERPROFILE%\Connect\logs`: As the name implies, log files are created and updated here. These log files are plain text files. Virus protection may have an impact on the speed of the whole Connect suite.
- `%USERPROFILE%\Connect\temp`: Storage folder for temporary data, usually intermittent files in multiple folders. Virus protection on this folder and its subfolders may have a serious impact on the performance of Connect.
- `%USERPROFILE%\Connect\workspace`: Usually containing settings and helper files and folders. Supervising this folder with a virus protection system should not have too much of an impact on the speed of the whole Connect suite.

Database 1

Depending on the components installed, a database instance is created in a folder called "**connect.database**" under the Windows system temp folder. This folder is accessible via the standardized system variable `%TMP%`. Usually, folders holding such temporary files and folders should be excluded from a virus protection, because this influences the overall performance of

the whole system at all. However the responsible person for the computer protection has to decide about the monitoring of such temporary folders following the company guidelines.

Database 2

Another database instance for Connect will be hold and used under the folder, which is intended to hold data, accessible by and for all users. The path to this folder is stored in the standardized system variable `%PROGRAMDATA%`. The Connect database instance is located in the subfolder "**Objectif Lune\OL Connect\MariaDB**".

As this database will be in extremely strong usage, virus protection on this folder and its sub-folders may have a **serious** impact on the performance of Connect.

Database Considerations

This page describes the different considerations and pre-requisites for the database back-end used by PrintShop Mail Connect, whether using the MariaDB instance provided by the installer, or pre-existing (*external*) instance.

Using the MariaDB Instance from the Installer

The MariaDB Instance provided in the "Installation Wizard" on page 32 is already pre-configured with options to provide the most stable back-end setup.

Installing Connect using an existing MySQL instance

If MySQL Server is already present and you wish to use it, the following should be taken into consideration:

- The minimum supported MySQL version is MySQL 5.6.
- The MySQL account must have access to all permissions using the GRANT Command, including creating databases.
- The database configuration must include the following options:
  - **max_connections = 200** : PrintShop Mail Connect uses *a lot* of database connections. This number ensures that even in high volume environments, enough connections will be available.

    - **max_allowed_packet = 500M** : In some implementations, especially when using Capture OnTheGo, large packet sizes are required to allow transferring binary files. This substantial packet size maximum setting ensures that the data received by PrintShop Mail Connect will be able to be stored within the database.
    - **character-set-server = utf8 , collation-server = utf8_unicode_ci , default-character-set=utf8** : These indicate database support for UTF-8/Unicode.
- The database configuration must allow the use of mixed case table names.
  This is particularly an issue on Linux MySQL installations.
- The SQL instance must be open to access from other computers. This means the bind-address option should not be set to 127.0.0.1 or localhost.

> **Warning**
>
> If you chose **not** to install the supplied MariaDB database, and instead opt for using a pre-existing (*External*) database then you yourself must ensure that the *External* database is accessible to Connect.
>
> Upland Objectif Lune will take no responsibility for setting up database connections to any but the supplied MariaDB database.

Options available within the installer:

- The Configuration page for the local MySQL is displayed.
- MySQL settings are pre-filled with default values if no existing MySQL database configuration is found.
- MySQL settings are pre-filled with existing database configuration settings, if they point to a MySQL database type.

Installing Connect using an existing Microsoft SQL Server instance

If Microsoft SQL Server is already present and you wish to use it, the following should be taken into consideration:

> **Warning**
>
> If you chose **not** to install the supplied MariaDB database, and instead opt for using a

pre-existing (*External*) database then you yourself must ensure that the *External* database is accessible to Connect.

Upland Objectif Lune will take no responsibility for setting up database connections to any but the supplied MariaDB database.

Note

Since PrintShop Mail Connect version 1.6 the minimum required version of the MS SQL Server is **SQL Server 2012**.

- When MS SQL is selected, the default values for root user are **sa** and **1433** for the port.
- If database settings from a previous OL Connect installation are found, the pre-exising settings will be displayed for the matching database type. For MS SQL settings, this will only work if they were created with Server Config Tool 1.5.0 or later, or the Installer for OL Connect 1.6.0 or later.
  If the database type is changed in the Installer configuration page, the default values for this database type will be displayed.
  If the pre-existing database settings are set to Hsqldb, the default database type selection will be MySQL.
- Selected database settings are stored in the preferences, and can be found in this file: *C:\ProgramData\Objectif Lune\OL Connect\.settings\ConnectHostScope\com.objectiflune.repository.eclipselink.generic.prefs*

When modifying Connect

- If the local MariaDB is removed from an installation, the Database Configuration page will offer additionally the **Microsoft SQL Server** database type with respective default values.
- If local MariaDB is added to an installation, the usual MariaDB Configuration page with default values will be displayed.

If the user has installed the Installer Supplied MySQL (2021.2 or earlier) or MariaDB (2022.1 and later) and then switches to an *external* Microsoft SQL by using the Server Configuration Tool, the supplied local database cannot be switched off. By design the installer adds a service dependency between Connect Server and the supplied MariaDB \ MySQL service.

To remove this dependency the user needs to do the following

1. Have a foreign Microsoft SQL running, ready for use with Connect Server.
2. Use the **Server Configuration Tool** "Database Connection preferences" on page 374 to switch the database to Microsoft SQL.
3. Re-start the Connect Server Service, so that the modifications become active.
4. Counter check that everything is working properly with Microsoft SQL.
5. Open a command-line prompt with full administration rights.
6. Enter the command `sc config OLConnect_Server depend= /`. This removes the dependency.
   Please be aware: The key word `depend` must be followed immediately by the equal sign, but between the equal sign and the forward slash there must be a space.
   Additional information can be found here: http://serverfault.com/questions/24821.
7. After the dependency has been removed, it is possible to stop the supplied MariaDB \ MySQL service (OLConnect_MySQL).

Environment considerations

Terminal Server/Service Support

PrintShop Mail Connect does not support Terminal Server (or Terminal Service) environment as possible under Windows 2000, 2003 and 2008. This is to say, if Terminal Service is installed on the server where PrintShop Mail Connect is located, unexpected behaviours may occur and will not be supported by Upland Objectif Lune. Furthermore, using PrintShop Mail Connect in a Terminal Service environment is an infringement of our End-User License Agreement.

Virtual Machine Support

PrintShop Mail Connect supports the following virtual environments:

- VMWare Environments. This includes VMWare Player, VMWare Workstation as well as VMWare ESX Server.
- VMWare VMotion. This means the virtual machine hosting PrintShop Mail Connect can be automatically moved from one ESX server to another in a clustered installation. Note that if all servers in a VMotion cluster are not strictly identical, you will have to provide the Objectif Lune Activations team with all possible magic numbers so they can generate a license that works on all servers. Obtaining the magic numbers is simply a

question of manually moving the VM on which Connect is installed to each server, and recording the magic number for each of them.

- Microsoft Hyper-V/Azure infrastructure environments.

PrintShop Mail Connect is *not* officially supported on any other virtual machines such as Virtual PC, Parallels, Bochs, Xen, etc.

> **Warning**
>
> Copying (duplicating) a Virtual Machine with Connect installed and using both images simultaneously constitutes an infringement of our End-User License Agreement.

> **Note**
>
> While some virtual machine environments (from VMWare and Microsoft) are supported, other virtual environments (such as Parallels, Xen and others) are not supported at this time.

Remote Desktop Support

Tests have demonstrated that PrintShop Mail Connect can be used through Remote Desktop. It is however possible that certain combination of OS could cause issues. If problems are encountered, please contact OL Support and we will investigate.

PrintShop Mail Connect 1.3 and later have been certified under Remote Desktop.

32-bit or 64-bit Operating Systems?

PrintShop Mail Connect is a 64-bit software and can ***only*** be installed on 64-bit operating systems.

Antivirus Considerations

- Antivirus software may slow down processing or cause issues if they are scanning in temporary folders or those used by PrintShop Mail Connect. Please see "Antivirus Exclusions" on page 13 for more information.

- Antivirus software might interfere with installation scripts, notably a VBS script to install fonts. McAfee, in particular, should be disabled temporarily during installation in order for MICR fonts to install and the installation to complete successfully.

Windows Search Indexing Service

Tests have concluded that the Windows Search service, used to provide indexing for Windows Search, can interfere with Connect when installing on a virtual machine. If the installation hangs during the last steps, it is necessary to completely disable this service during installation.

- Click on Start, Run.
- Type in **services.msc** and click OK.
- Locate the **Windows Search** service and double-click on it.
- Change the **Startup Type** to **Disable**, and click **Stop** to stop the service.
- Try the installation again.
- Once complete, you may re-enable the service and start it.

Commandline switches and .ini entries

PrintShop Mail Connect is intended to work stably and reliably, based on Java and the Eclipse framework. To ensure this reliability and robustness, many Java and Eclipse parameters have been tested and tuned, which is reflected in the respective .ini entries and the used command line switches. A collection of valuable settings has been elaborated and found its entry in PrintShop Mail Connect “good switches list” (called the “whitelist”).

The protection of the end user’s system is one of our main goals and therefore we have implemented a very strict verification mechanism, which ensures, that only these whitelisted ini entries and command-line switches are accepted, when one of Connect components is started and run. Please be therefore advised, that any non-whitelisted ini entry or command-line switch will be accepted and will - if tried to be used - lead to the respective application’s “sudden death”. If you should encounter such a behaviour then please double-check your Connect log file/s for respective entries.

Language and Encoding Considerations

Please note the following considerations:

- **Language**:
  - PrintShop Mail Connect is currently offered in several languages. These languages can be switch between via the Preferences dialog. The current languages include:
    - English
    - French
    - German
    - Spanish
    - Italian
    - Korean
    - Portuguese
    - Chinese (Simplified)
    - Chinese (Traditional)
    - Japanese.

    The default language is English.
    The PrintShop Mail Connect help system (this document and the online help) is currently only available in English and (for the biggest part) in French.
- **Encoding**:
  - Issues can sometimes be encountered in menus and templates when running PrintShop Mail Connect on a non-English operating system. These are due to encoding issues and will be addressed in a later release.

Network Considerations

The following should be taken into consideration in regards to network settings and communications

- If a local proxy is configured (in the **Internet Explorer Options** dialog), the option **Bypass proxy server for local addresses** must be checked, or some features depending on local communication will not work.

Firewall/Port considerations

The following describes all of the ports that can be used by an OL Connect solution. IT staff may decide the firewall strategy to follow for their internal requirements and needs depending on the statements outlined herein.

	Listens on port #	Destination port #	Type	Comment
Messenger	**5863/5864**	**5863/5864**	TCP+UDP	Used for inter-module communication
Connect Server	**9340**		TCP	Used for Connect REST API
Connect Server	**9350**		TCP	Dedicated internal connection for inter-process communication (i.e. between engines)
HTTP Server	**8080**		TCP	
HTTPS Server	**443**		TCP	
NodeJS Server	**9090**		TCP	
NodeJS HTTPS Server	**8443**		TCP	
SMTP Input plugin	**25**, or **587** when encryption is enabled		TCP	
Email Input plugin		110	TCP	Default POP3 port
Secure Email Input plugin		**993**	TCP	
Send Email plugin		**25**	TCP	Default SMTP port
Secure Email Output plugin		**587**	TCP	
LPR		515	TCP	
LPD	515		TCP	
Telnet	**9100**		TCP	
FTP Input/Output		**21**	TCP	
MySQL / MariaDB	**3306**		TCP+UDP	
Microsoft SQL Server	**1433**		TCP+UDP	
HyperSQL	**9001**		TCP	

- Port numbers in **bold** type are user configurable.
- Port numbers in **bold underlined** type are based on the type of database used.
- Some of the ports listed above may also be used by other modules.

- User-configurable modules may use other ports entirely, depending on the settings defined by the end user. A few examples:
  The **Create Email Content** plugin will use the port defined in the Mail host setting in Workflow's OL Connect Preferences.
  The ports used by the **HTTP Client Input** task, **Legacy SOAP Client** and **SOAP Client** plugin depend on the configured URL.

Performance Considerations

This page is a guide to getting the most performance out of PrintShop Mail Connect as well as a rough guideline to indicate when it's best to upgrade.

Performance analysis details

In PrintShop Mail Connect, after each print job, a message dialog displays a print job summary (see Print Job Summary dialog). This summary may help you determine how best to maximize the performance of PrintShop Mail Connect.

With no Performance Pack, PrintShop Mail Connect's Weaver engine can generate output at 250 **PPM** (pages, or emails per minute) for Subscription licence, or 1,000 PPM for Perpetual licence. Additional Performance Packs increase this quota.

If your print jobs are not running at the licensed speed, there may be several ways to improve the performance, as described below. Note however that it is not guaranteed that the licensed speed can be achieved with **any** job. Creating output for templates with very complex scripts or complex graphics resources will take a certain amount of time, even on high-end hardware.

If your print jobs are running at the licensed speed, then improving performance will require purchasing either a Performance Pack or upgrading to PlanetPress or PReS Connect (see Performance Packs). For advice please contact your local sales office (see Objectif Lune's Contact page).

Engine configuration

A **Merge engine** merges the template and the data to create Email output, or to create an intermediary file for Printed output. The intermediary file is in turn used by a **Weaver engine** to prepare the Print output.

Configuring these engines to match the hardware configuration is probably the fastest and most effective way to improve Connect's performance. See "Engines preferences" on page 84 for how to do this.

Template optimization

When you find that the speed per Merge engine - the Content Creation speed - is low, optimizing a template can make a huge difference. For advice on how to optimize a template see: "Optimizing a template" on page 1026.

Network and internet connections

Use a fast network and internet connection or avoid loading external or internet resources. Using images, JavaScript or CSS resources located on a slow network or on a slow internet connection will obviously lead to a loss of speed. While we do our best for caching, a document with 5,000 records which queries a page that takes 1 second to return a different image each time will, naturally, slow output generation down by up to 83 minutes.

Hardware configuration

When processing speed is important, the following is suggested before looking into Performance Packs to enhance performance (and after addressing the other issues mentioned in this topic).

- **Antivirus exclusions**. Sometimes, virus scanners, other security software or indexing services can interfere. It can help to disable those kinds of tools for the areas where Connect stores intermediate files. You could exclude the entire `C:\Users\<connectuser>\Connect` folder. See also: "Antivirus Exclusions" on page 13.
- Use a **high-performance, low-latency hard drive**. Connect benefits from fast I/O. Preferably use a Solid State Drive (SSD) or similar for storage.
- Use at least **8+ GB High-Quality RAM**. Check memory usage while the Print command is being executed to see if you need more than the minimum of 8GB. Assuming that the Designer needs 1GB each, and that each engine needs 1GB as well, you can roughly estimate how much memory is needed.
- **Consider using a physical machine** instead of a virtual machine. When running on a Virtual Machine, the machine may report that it has sufficient hardware (cores) available, but in a virtual environment you need to make sure that this hardware is not being shared with lots of other virtual machines.

- Consider using hardware with more **physical cores**. PrintShop Mail Connect doesn't limit the number of Merge engines that is used for a Print job, so if the number of physical cores is low, it makes sense to see if that can be increased. When running on a virtual machine, this is usually easy. When running on a physical machine, it means that you may have to switch hardware.
- For both virtual and non-virtual environments, make sure the machine is not busy with all kinds of other processes.

System requirements

These are the system requirements for PrintShop Mail Connect 2022.1

Operating system (64-bit only)

- Microsoft Windows 11
- Microsoft Windows 10 (Pro and Enterprise versions only)
- Microsoft Windows 8.1
- Microsoft Windows Server 2022
- Microsoft Windows Server 2019
- Microsoft Windows Server 2016
- Microsoft Windows Server 2012/2012 R2

> **Note**
>
> PrintShop Mail Connect 2022.1 is *expected* to run on some older operating systems, but just as Microsoft no longer supports these older operating systems, Upland Objectif Lune will **not** provide support for PrintShop Mail Connect products running on them.
>
> The historic operating systems that it is *expected* PrintShop Mail Connect 2022.1 will continue to run on include: Microsoft Windows 7; Microsoft Windows Server 2003; and Microsoft Windows Server 2008 R2.

Virtual Environments

- VMWare/VSphere
- Hyper-V (8.0)

- Azure
- Amazon Web Services (AWS). Note that only EC2 M4 was certified, other instances may not work as expected.

Minimum hardware requirements

As with any software application, minimum hardware requirements represent the most basic hardware on which the software will run. Note however that settling for the minimum specification is unlikely to produce the performance you expect from the system. It can be used when configuring a trial or a development system, however.

- File system: NTFS (FAT32 is not supported)
- CPU: multi-core
- RAM: 6GB
- Disk Space: 4GB for the software modules, 5GB for work files*

* This depends on the amount of data you process through OL Connect. For instance, a PostScript file containing several thousands of documents could easily take up several GBs.

Recommended hardware requirements

Due to its versatility, OL Connect is used for a wide variety of applications. Consequently, it is difficult to determine which hardware configuration will produce the best results for any given implementation. The following specs should therefore be viewed as a general guideline that is most likely to produce expected results for most implementations. You should, however, keep in mind that it may not represent the optimal setup for your particular application.

For more information and tips about performance considerations, see "Performance Considerations" on page 23.

- File system: NTFS (FAT32 is not supported)
- CPU: Intel Core i7-4770 Haswell or better
- RAM: 16GB
- Disk Space: 4GB for the software modules, 20GB for work files*[1]
- Storage Type: Solid State Drive (SSD)
- Networking: 10Gb Ethernet

*1 This requirement depends upon the amount of data you process through OL Connect. For instance, a PostScript file containing several thousands of documents could easily take up several GBs.

> **Note**
>
> As with any JAVA application, the more RAM that is available, the faster PrintShop Mail Connect will execute.

Requirements for individual Connect modules

OL Connect Products comprises multiple modules that can be operated separately on multiple PCs. Each module has its own set of requirements that may differ from the other modules. While the hardware requirements described above are relatively generic when installing all Connect modules on a single server, they should not be interpreted literally for each individual module.

When installing on multiple PCs, keep the following rules of thumb in mind:

- The Connect Workflow module requires less RAM but fast hard drive access. It also benefits from fast multi-core CPUs, in order to run processes in parallel.
- The Connect Server module requires more RAM and benefits from fast multi-core CPUs. Disk access speed is less of a concern.
- The Connect Designer module requires more RAM and fast disk access to provide a responsive user-experience.
- The back-end database (MariaDB by default) benefits from more RAM, speedy disk access and fast networking as it will be solicited by all modules simultaneously.

> **Note**
>
> As with any JAVA application, the more RAM that is available, the faster PrintShop Mail Connect will execute.

Editions of Connect Products

There are three editions of OL Connect: PrintShop Mail Connect, PlanetPress Connect and PReS Connect.
While all three editions share common modules, they are generally not used for the same

purposes. Technically speaking, their hardware requirements would therefore be the same but in practice, PReS Connect is likely to require higher-end hardware while PrintShop Mail Connect will generally require less power to achieve expected results.

Installation and Activation

This topic provides detailed information about the installation and activation of PrintShop Mail Connect 2022.1.

> **Note**
>
> A PDF version of this guide is available for use in offline installations. Click here to download it.

Where to obtain the installers

The installer for PrintShop Mail Connect 2022.1 can be obtained on DVD or downloaded as follows:

- If you are a **Customer**, the installer can be downloaded from the Objectif Lune Web Activation Manager (https://www.objectiflune.com/webactivationmanager/) or through the OL Update Manager if it is activated.
- If you are a **Reseller**, the installer can be downloaded from the Objectif Lune Partner Portal site (https://extranet.objectiflune.com/) or through the OL Update Manager if it is activated.

Installation - important information

For important information about the Installation, including requirements and best practices, please see the following topics:

- "Installation prerequisites" on the next page
- "User accounts and security" on the next page
- "Migrating to a new workstation" on page 48
- "Upgrading from previous Connect versions" on page 54

Installation - "How to" guides

For information on how to conduct the installation itself, choose from the following topics:

- "Installation Wizard" on page 32
- "Running Connect installer in Silent Mode" on page 38
- "Installing PrintShop Mail Connect on Machines without Internet Access" on the facing page

Activation

For information on licensing, please see "Activating a License" on page 45.

Installation prerequisites

- Make sure your system meets the "System requirements" on page 25.
- PrintShop Mail Connect Version 2022.1 can be installed under a regular user account with Administrator privileges., see "User accounts and security" below.
- PrintShop Mail Connect **must** be installed on an NTFS file system.
- PrintShop Mail Connect requires **Microsoft .NET Framework 4.5** already be installed on the target system.
- Connect 2019.1 requires updated Connect License and/or Update Manager.
  See "Upgrading from previous Connect versions" on page 54 for details.

User accounts and security

Windows user account

Connect requires local Windows Administrator rights when installing the software and activating the software license. This is to allow read/write access to protected Windows folders and registry entries.

Once installed Connect requires only standard Windows user credentials to run.

The following links contain the details as to when and where Windows Administrator rights are required:

- Connect Installation: "Installation Wizard" on page 32
- Activating Connect: "Activating PrintShop Mail Connect" on page 47

Permissions for PrintShop Mail Connect Designer

PrintShop Mail Connect Designer does not require any special permissions to run besides that of a regular program.

It does not require administrative rights and only needs permission to read/write in any folder where templates are located.

If generating Print output, PrintShop Mail Connect Designer requires permission on the printer or printer queue to send files.

Installing PrintShop Mail Connect on Machines without Internet Access

Installing PrintShop Mail Connect2022.1 in offline mode requires some extra steps. These are listed below.

Updating Connect

Updating to Connect 2019.1 from earlier Connect version

In order to update PrintShop Mail Connect to 2019.1 it is first necessary to update the Connect License.
For details on how to upgrade the Connect License offline see the **Upgrading Connect on machines with no internet access** section in the document.

Initial Connect Installation

GoDaddy Root Certificate Authority needs to be installed

In order to install PrintShop Mail Connect it is necessary for the GoDaddy Root Certificate Authority to be installed (G2 Certificate) on the host machine and for this to be verified online. When a machine hosting the installation does not have access to the Internet, the installation will fail because the verification cannot be performed. To solve this problem one must first ensure that all Windows updates have been installed on the host machine. Once the Windows updates are confirmed as being up to date, then complete the following steps:

1. Go to https://certs.godaddy.com/repository and download the following two certificates to copy to the offline machine:
    - GoDaddy Class 2 Certification Authority Root Certificate - G2 - the file is gdroot-g2.crt
    - GoDaddy Secure Server Certificate (Intermediate Certificate) - G2 - the file is gdig2.crt
2. Install the certificates: Right mouse click -> Install Certificate, and follow the steps through the subsequent wizard.
3. Now copy the PrintShop Mail Connect installer to the offline machine and start the installation as normal

Windows certificate validation - Certificate Revocation List retrieval should be switched off

For your security Objectif Lune digitally signs all relevant files with our own name and certificate. The integrity of these files is checked at various times by different, context related, methods. One of these checks, done during the installation process, uses the Windows certificate validation check. .

The Windows certificate validation process not only checks the integrity of a file against its signature, but also usually checks if the certificate itself is still valid. That check is done against the current Certificate Revocation List (CRL), which needs to be retrieved from the internet. However, if the machine in question does not have internet access, the retrieval of the CRL must fail, which will lead to subsequent validation issues.

To circumvent such issues it is **highly recommended** to switch off the CRL retrieval prior to installing Connect on machines without internet access. There is no security risk associated with this, as the CRLs would never be retrievable without internet access, anyway. Advantage of the switch will not only be found during the installation and operation of Connect, but also in some speed improvements for any application which use signed binaries.

To switch off CRL retrieval on the computer, complete the following steps:

1. Open the “Internet Options” via the Control Panel
2. Select the “Advanced” tab and scroll down to “Security” node.
3. Uncheck the entry “Check for publisher’s certificate revocation” under that node.
4. Click the OK button to close the dialog.
5. Re-start the computer.

Updating from Connect versions predating 2019.1

In order to update PrintShop Mail Connect to 2022.1 from Connect versions prior to 2019.1 it is first necessary to update the Connect License.
For details on how to upgrade the Connect License see "Users of Connect prior to 2019.1" on page 54

Starting the PrintShop Mail Connect installer

The PrintShop Mail Connect installer is supplied as an executable file.

Double click on the executable file and after a short pause the Setup Wizard will appear to guide through the installation steps.

> **Note**
>
> Note that PrintShop Mail Connect **requires** prior installation of Microsoft .NET Framework 4.5. For a full list of other prerequisites, see "Installation prerequisites" on page 29.

Running the Installation with extra logging

The installer can be run with enhanced logging options, if needed.
To do so, run the PrintShop Mail_Connect_Setup_x64.exe from the command line with one of the following command line options:

- `PrintShop Mail_Connect_Setup_x64.exe --verbose`
  This adds extra debugging style logging to the installation process.
- `PrintShop Mail_Connect_Setup_x64.exe --trace`
  This adds full trace style logging to the installation process. The log file this produces will be very large, as this option logs everything.

Prerequisites Installation

The PrintShop Mail installer will check for prerequisite technologies as the first step in the installation process. If this check finds some technologies are missing, it will install those technologies, before continuing with the installation.

Welcome screen

After any prerequisites are installed, the PrintShop Mail installer Welcome screen appears. Click **Next** to continue with the PrintShop Mail installation.

If the installation is an upgrade over a pre-existing Connect installation, the installer will first uninstall the earlier version.

If you would like to retain the usage information from that pre-existing Connect installation, do not select the **Remove User data** checkbox option.

For information about exactly what data would be saved or deleted, please see "Upgrading from Connect 2021.2 or earlier with Remove User Data CHECKED:" on page 57.

License Agreement

The next page displays the Skin/Formats/CrossReferencePrintFormat("" on page 1), which needs to be read and accepted before clicking **Next**.

Component Selection

After clicking the Next button, the Component Selection page appears, in which the different components of PrintShop Mail Connect can be selected for installation.

The options are:

- **Base**: The installation files required for any PrintShop Mail Connect Connect installation. This component is not optional.
- **Designer**: The Designer module (see "The Designer" on page 107) can be installed standalone (with no other installed modules) on as many machines as you like. It does not require a license to run as a standalone designer tool . This allows any number of people to use the Designer for creating jobs, but without production capabilities such as automation and commingling. The Designer module is optional, but it is recommended that it always be installed.
- **MariaDB Server**: A supplied MariaDB database used by PrintShop Mail Connect. The database is used for referencing temporary Connect files and for sorting temporarily extracted data, and similar.

  A pre-existing MariaDB, MySQL or Microsoft SQL server (referred to as an ***external*** database, in this documentation) *could* be used instead, for the same purposes. The

external database could reside on the same computer or on a separate server.
If you wish to make use of an *external* database, please make sure the **MariaDB** option is not selected.

> **Warning**
>
> If you chose **not** to install the supplied MariaDB database, and instead opt for using a pre-existing (*External*) database then you yourself must ensure that the *External* database is accessible to Connect.
>
> Upland Objectif Lune will take no responsibility for setting up database connections to any but the supplied MariaDB database.
>
> See "Database Considerations" on page 15 for more information about setting up *external* databases.

- **Destination folder**: This is the location where Connect components are to be installed.

  Use the Browse button to navigate to a folder other than the default, if required.

> **Note**
>
> The installation path cannot contain any non ASCII characters (such as Asian language Unicode characters). Nor can it contain characters that Windows disallows in filenames (such as '?', '>' or trailing spaces).
> If an invalid character is entered, the Installation Path entry box will turn red and a description of the error will be displayed in the information area.

The installer calculates how much disk space is required for installing the selected components, along with how much space is available.

- **Total Required Space** : Displays the amount of disk space required for the selected components.
- **Space Remaining**: Displays the amount of space available after installation on the drive currently in the Installation Path.

Database Configuration

The Default **Database Configuration** page appears if the supplied *MariaDB* module was selected for installation in the *Product Selection* screen. It defines the administrative root password for the MariaDB server as well as which port it uses for communication.

The installer will automatically configure the Connect *Server* to use the supplied password and port.

- **Port**: The port on which MariaDB will expect requests to come though, and through which it itself responds.
  A check is run to confirm whether the specified TCP\IP Port Number is available on the local machine. If it is already being used by another service (generally, an existing MySQL or MariaDB installation), the number is highlighted in red and a warning message is displayed.

  > **Note**
  >
  > The MariaDB database controlled by the *OLConnect_MariaDB* service communicates through port 3306 by default.

- **Root password**: Enter the password for the 'root', or administration account, for the MariaDB server.
  Use the eye icon to toggle between displaying or masking the password entry.
  We recommend that the password be at least 8 characters long and contain at least one of each of the following, even though password selection strength is not enforced by the installer:
  - a lower case character (a, b, c ...)
  - an upper case character (A, B, C ...)
  - a numeric digit (1, 2, 3 ...)
  - a punctuation character (@, $, ~ ...)

  For example: "This1s@K"

> **Note**
>
> When updating from an earlier Connect version, the appropriate MariaDB password **must** be entered or the update will fail.
>
> If the password is subsequently forgotten, then MariaDB must be uninstalled and its database deleted from disk before attempting to reinstall.

- **Allow remote client access** checkbox: Click to enable external access to the MariaDB server.

> **Note**
>
> This option is required if MariaDB Server will need to be accessed from any other machine.
>
> It will also be required if the MariaDB database is on a separate machine to this PrintShop Mail Connect installation.

> **Tip**
>
> This option may represent a security risk if the machine is open to the internet.
>
> We heavily recommended that your firewall is set to block access to port 3306 from external requests.

- **Username**: Enter the MariaDB user name that will be associated with OL Connect. The default username for new installations is **olconnect**.
- **Password**: The password associated with the selected user.

  Use the eye icon to toggle between displaying or masking the password entry. The password is not validated for password strength, so any entry is acceptable.

Ready to install

This page confirms the installation selections made.

Click **Install** to start the installation itself. This process can take several minutes.

Installation Finished

This screen describes a summary of the components that have been installed.

- **Configure update checks** checkbox: This option is enabled by default. It causes the **Product Update Manager** to run after the installation is complete. This allows configuring PrintShop Mail Connect to regularly check for entitled updates.
  Note: this checkbox may not be available in the event that an issue was encountered during the installation.

When ready, click the **Finish** button to close the installation wizard, and initialize the Product Update Manager, if it was selected.

The Product Update Manager

If the **Configure Update Check** option has been selected, a message will be displayed after clicking "*Finish*" in the setup. The message details the information that needs to be sent back to Upland Objectif Lune in order to determine when/if the software needs updating.

Click "Yes" to install or open the Product Update Manager where the frequency with which the updates can be checked and a proxy server (if required) can be specified.

> **Note**
>
> If the Product Update Manager was already installed by another Upland Objectif Lune application, it will be updated to the latest version and will retain the settings previously specified.

Select the desired options and then click **OK** to query the server and obtain a list of any updates that are available for your software.

- Note that the Product Update Manager can also be called from the "***Objectif Lune Update Manager***" option in the Start menu.
- It can be uninstalled via Control Panel | Programs | Programs and Features.

Product Activation

After installation, it is necessary to activate the software. See "Activating a License" on page 45 for more information.

Before activating the software, please wait 5 minutes for the database to initialize. If the software is activated and the services rebooted too quickly, the database can become corrupted and require a re-installation.

Running Connect installer in Silent Mode

Updating from Connect versions predating 2019.1

In order to update PrintShop Mail Connect to 2022.1 from Connect versions prior to 2019.1 it is first necessary to update the Connect License.
For details on how to upgrade the Connect License see "Users of Connect prior to 2019.1" on page 54

General information

PrintShop Mail Connect can be installed from the command line in "*silent mode*" to allow for scenarios such automated installations during company wide roll-outs, or to allow for unattended out-of-hours updates. The trigger for the Connect Installer to run in silent mode is a text file with the fixed name **installProperties.ini** located in the same folder as the PrintShop Mail Connect installation executable file.

How to prepare the **installProperties.ini** file is detailed in the following sections.

Installation Properties file

The basic rules for the **installProperties.ini** file are:

- Comment Lines start with the semi-colon character ';'
  Example: `; The options to configure an external database`
- The properties are listed in **Key = Value** pair format.
  Example: `product.ServerExtension = false`
- Quotes are used for path strings, but not for other string types
  Example: `path = "c:\Program Files\Objectif Lune\OL Connect"`
- Property values are case insensitive

The installation settings fall into three distinct categories:

- "[Logging]" below for setting the installation logging options
- "[Installation]" on the facing page for selecting what gets installed
- "[Uninstall]" on page 42 handles properties related to product Uninstallation. These also impact Maintenance mode.

[Logging]

This section handles **Silent Installer** logging options.

The logging Key pairs are as follows:

- **verbose**: Boolean (Default: `false`)
  A basic log is always created by the installer, with or without the verbose option. The verbose option is more suitable for debugging purpose.
  If set to true, then a verbose log file is created in the logging path specified in the INI file.
  If no logging path is specified in the INI file, then the default one is used.
  If set to false, standard logging is done.
- **path**: String (Default: `%PROGRAMDATA%\Objectif Lune\Installation Logs`)
  Sets the folder to which the installation log will be written.
  Only the log folder should be specified here, not the log file name.

> **Note**
>
> The log file name's format is set automatically and uses the format *Installer-YYYY-MM-DD-####.#.#.log*, where:
>
> - *YYYY-MM-DD* = The date the log was created
> - ####.#.#= The PrintShop Mail Connect version number
>
> The file name for maintenance installation begins with "Maintenance" rather than "Installer"

Example:

```
; Logging properties
[Logging]
verbose = false
path = "c:\temp\Silent Install"
```

[Installation]

This section handles various installation parameters as well as product selection.

The logging Key pairs are as follows:

- **product.<name>**: Boolean (Default: `false`)
  Each OL Connect product has its own entry, which can be set to true (to install) or false (to omit from installation).
  What products are available for installation is determined by which OL Connect branding is being installed. The products for PrintShop Mail Connect branding are as follows:
  - **product.Designer**
  - **product.PrintManager**
  - **product.MariaDB**
- **path**: String (Default: `%PROGRAMFILES%\Objectif Lune\OL Connect`)
  Sets the installation root folder for the PrintShop Mail Connect applications.
- **database.configure**: Boolean (Default: `true`)
  If set to "false", then the database configuration is skipped.

  > **Note**
  >
  > If database configuration is skipped (`database.configure = false`), then none of the `database.xxx` properties below are required, and these properties will be ignored, even in they are included in the INI file.

- **database.system**: String, Optional (Default: there is no default for this setting)
  Entry needs to be from one of the following options: **mariadb**, **mysql**, **mssqlserver**

  > **Note**
  >
  > If `product.MariaDB = true` has been set:
  >
  > - This setting is optional. Otherwise it is required.
  > - The value is *required* to be `mariadb`, if the value is to be provided.

- **database.host**: String, Optional (Default: there is no default for this setting)
  If `product.MariaDB = True` has been set, this value is required to be set to `localhost`, if provided.
- **database.port**: Numeric, Optional (Default: default is MariaDB's default port, 3306)
  The database engine port.
  The required entry depends upon the selection made in `database.system`.
  If `product.MariaDB = True`, then the port should be set to 3306.
- **database.rootpassword**: String (Default: there is no default for this setting)
  Database root password.
  There is no default value, and if this is left unspecified the installation will fail.
- **database.username**: String (Default: `olconnect`)
  The username that PrintShop Mail Connect will use to connect to the database.
- **database.password**: String (Default: there is no default for this setting)
  The password that OL Connect will use to connect to the database.
  There is no default value, so if this is left unspecified, then the installation will fail.
- **database.instance**: String, Optional (Default: there is no default for this setting)
  Only valid if `database.system = mssqlserver`.
- **database.schema**: String (Default: there is no default for this setting)
  Specifies the database schema to use. Required. (Optional if `product.MariaDB = true`)
- **database.encryptedconnection**: Boolean (Default: `False`)
  Specifies the database schema to use.
  (Optional if if `product.MariaDB = true`)
- **server.connection.configure**: Boolean, Optional (Default: `True`)
  If set to `False`, then the server connection configuration is skipped.
- **server.connection.user**: String (Default: `olc-user`)
  The server connection username.
- **server.connection.password**: String (Default: there is no default for this setting)
  The server password. Required if `server.connection.configure = true`.
- **server.connection.port**: Numeric (Default: 9340)
  The server port number.

Example:

```
; Installation settings
[Installation]
product.Designer = true
product.Server = true
product.PrintManager = true
product.ServerExtension = false
```

```
product.MariaDB = true
product.Messenger = true
RegisterService.connectServer = true
server.username = Administrator
server.password = ObjLune
server.connection.user = olc-user
server.connection.password = secret
database.rootpassword = @Admin2022
database.username = olconnect
database.password = @Admin2022
database.remoteaccess = true
path = "c:\Program Files\Objectif Lune\OL Connect"
```

[Uninstall]

This section handles properties related to Uninstallation, but also Maintenance mode installations.
These options have no effect at all if PrintShop Mail Connect is not present on the system.

The logging Key pairs are as follows:

- **remove**: Boolean (Default: `False`)
  If the present version of the installer is already installed, this property defines the installer behaviour.
  The installer will perform a *Remove MSI* action on OL Connect if `remove = true`, whilst a *Modify MSI* action will be performed if `remove = false`.

  Notes:

  - This option has no effect if the product is not installed on the current system
  - If the *Remove* action is taken, then the equivalent of an uninstall is done, while a *Modify* action will change the components installed on the system based on the ones defined in the INI file [Installation] section, allowing removal or addition of components in the current installation.
- **keepdata**: Boolean (Default: `True`)
  Allows the user to specify if they wish to keep or remove user data (located under *%PROGRAMDATA%\Objectif Lune\OL Connect* when performing a product uninstall.

Example:

```
; Uninstallation/Repair properties
[Uninstall]
```

```
remove = true
keepdata = true
```

Properties file examples

Simple Connect installation example

Here is an example of a complete **installProperties.ini** file for a relatively simple PrintShop Mail Connect installation.

```
; OL Connect silent installer properties

; Logging properties
[Logging]
verbose = false
path = "c:\ProgramData\Objectif Lune\Installation Logs"

; Installation settings
[Installation]
product.Designer = true
product.Server = true
product.PrintManager = true
product.ServerExtension = false
product.MariaDB = true
product.Messenger = true
RegisterService.connectServer = true
server.username = Administrator
server.password = ObjLune
server.connection.user = olc-user
server.connection.password = secret
database.rootpassword = @Admin2022
database.username = olconnect
database.password = @Admin2022
database.remoteaccess = true
path = "c:\Program Files\Objectif Lune\OL Connect"

; Optional uninstallation settings
[Uninstall]
remove = true
keepdata = false
```

Exit Codes

Success

- 0 = Installation completed successfully / no specific error code was returned.

Pre-Installation Check (200s)

- 201: Operating system is not 64-bit
- 202: Windows Operating System is too old (pre-Windows 7)
- 203: Minimal UAC prerequisites were not met
- 204: Installation user did not have administrator rights

Silent installation properties (300s)

- 301: MariaDB product was selected, but no root password was supplied
- 302: MariaDB product was selected, but no user password was supplied
- 303: User provided database is to be used, but `database.system` property was missing
- 304: User provided database is to be used, but `database.system` property had invalid value
- 305: User provided database is to be used, but `database.host` property was missing.
- 306: database.instance property was used with a database system other than MS SQL Server
- 307: User provided database is to be used, but `database.username` property was missing.
- 308: User provided database is to be used, but `database.password` property was missing.
- 309: Connect server / server extension product was selected, but `server.connection.password` property was missing
- 310: Connect server extension product was selected, but `server.connection.host` property was missing

Upgrade errors (400s)

- 401: Some Connect applications were running and need to be closed before installation can proceed.

- 402: The installer brand does not match the brand of the OL Connect version currently installed. (PlanetPress)
- 403: The installer brand does not match the brand of the OL Connect version currently installed. (PReS)
- 404: The installer brand does not match the brand of the OL Connect version currently installed. (Printshop Mail)

License file validation (500s)

- 501: PrintShop Mail Connect license file is in older format
- 502: License Care Date does not allow installation of product
- 503: License brand mismatch with installer brand

Destination and selected product check (600s)

- 601: Server and Server Extension both were selected to be installed.
  Only one of the two may be installed on any one system.
- 602: No component was selected to be installed
- 603: Destination folder is invalid or there is too little disk space available

Installation aborted (700s)

- 701 - 725: Installation was aborted due to user cancellation

Service stop error (800s)

- 801: OLConnect_MySQL/OLConnect_MariaDB service could not be stopped during uninstallation

Activating a License

PrintShop Mail Connect comes with a 30 day trial license period during which time it is not necessary to have a commercial license to run the application.

This allows time for reviewing the applications and for organizing a commercial license. If a modification to the trail license is required, such as to allow an extension to the trial period, or for extra functionality, then a new activation code will need to be requested.

Obtaining the PrintShop Mail Connect Magic Number

To obtain an activation file the OL™ **Magic Number** must first be retrieved. The Magic Number is a machine-specific code that is generated based on the computer's hardware and software using a top-secret Objectif Lune family recipe. Each physical computer or virtual computer has a different Magic Number, and each requires a separate license file to be functional.

To get the PrintShop Mail Connect**Magic Number** open the **Connect Software Activation** application.

- Open the **Start Menu**
- Click on **All Programs** and browse to the **Objectif Lune** folder.
- Open the **Connect Software Activation** shortcut.
- The **PrintShop Mail Connect Software Activation** application consists of the following:
  - **License Information** subsection:
    - **Magic Number**: Displays the PrintShop Mail Connect Magic Number.
    - **Copy the magic number to the clipboard**: Click to copy the Magic Number to the clipboard. It can then be pasted in the activation request email using the Windows CTRL+V keyboard shortcut.
  - **Licensed Products** subsection:
    - **Name**: Displays the name of the application or module relevant to this activation.

      The Information button i provides detailed information about the application or module license.
    - **Serial Number**: Displays the trial license serial number or the activation serial number if the product has been activated in the past.
    - **Expiration Date**: Displays the date when the activation will expire, or the current date if the product is not activated.
  - **End-User License Agreement** - *Appears only when loading a license file*:
    - **License**: This box displays the EULA. Please note that this agreement is legally binding.
    - **I agree**: Select to accept the EULA. This option **must** be selected to install the license.
    - **I don't agree**: Select if you do not accept the EULA. You cannot install the license if this option is selected.

- **Load License File**: Click to browse to the Connect license file (**.olconnectlicense**), once it has been received.
- **Install License** - *Active only when a license file is Loaded*: Click to install the license and activate the software.
- **Close**: Click to cancel this dialog.
  Even if a license file has been Loaded, it will not be installed if this dialog is Cancelled before the **Install License** button was clicked.

Requesting a license

After getting the Magic Number, a license request must be done for PrintShop Mail Connect:

- **Customers** must submit their Magic Number and serial number to Objectif Lune via the Web Activations page: http://www.objectiflune.com/activations. The OL Customer Care team will then send the PrintShop Mail Connect license file via email.
- **Resellers** can create an evaluation license via the Objectif Lune Partner Portal by following the instructions there: http://extranet.objectiflune.com/

Note that if you do not have a serial number, one will be issued to you by the OL Activations team.

Accepting the license will activate it, after which the PrintShop Mail Connect services will need to be restarted. Note that in some case the service may not restart on its own. To resolve this issue, restart the computer, or start the service manually from the computer's Control Panel.

Activating PrintShop Mail Connect

To activate PrintShop Mail Connect, simply save the license file somewhere on your computer where you can easily find it, such as on your desktop. You can then load the license by double-clicking on it, or through the **PrintShop Mail Connect Software Activation** tool.

Activating the PrintShop Mail Connect license requires the user to have local Windows Administration rights.

- Using a user profile that has local Windows Administration rights, open the **Start Menu**
- Click on **All Programs**, then browse to the **Objectif Lune** folder.
- Run the "**Connect Software Activation**" tool.
- Click the **Load License File** button, and browse for the .olconnectlicense file you received from Upland Objectif Lune.

- Read the EULA and click the *I agree* option to accept it.
- Click **Install License** to activate the license. The license will then be registered on the computer and you will be able to start using the software.

> **Warning**
>
> After installation message will appear warning that the Server services will need to be restarted. Just click OK to proceed.

Migrating to a new workstation

The purpose of this document is to provide a strategy for transferring a OL Connect installation to a new workstation.

Before installing the software

Before upgrading to a new version, even on a new workstation, consult the product's release note to find out about new features, bug fixes, system requirements, known issues and much more. Simply go to the product page and look for the "**Release notes**" in the Downloads area.

You should also consult the following pages for some technical considerations before installing:

- "Network Considerations" on page 21
- "Database Considerations" on page 15
- "Environment considerations" on page 18
- "Installation prerequisites" on page 29
- "Antivirus Exclusions" on page 13

Downloading and installing the software

In order to migrate to a new workstation, the software must already be installed on the new workstation. Follow the "Installation and Activation" on page 28 guide to download and install the newest version of PrintShop Mail Connect on the new workstation.

Backing up files from the current workstation

The first step in migrating to a new workstation would be to make sure all necessary production files and resources are backed up and copied over to the new system.

> **Note**
>
> Although it is not necessary to convert all of your documents when upgrading to the latest version, we strongly recommended doing so. It is considered "Best Practice" to convert the documents to the version installed.

Backing up Connect Resources

The following resources are used by Connect and can be backed up from their respective folders:

- **Job Presets** (.OL-jobpreset):
  `C:\Users\[UserName]\Connect\workspace\configurations\JobCreationConfig`
- **Output Presets** (.OL-outputpreset):
  `C:\Users\[UserName]\Connect\workspace\configurations\PrinterDefinitionConfig`
- **OL Connect Print Manager Configuration files** (.OL-ipdsprinter):
  `C:\Users\[UserName]\Connect\workspace\configurations\PrinterConfig`
- **OL Printer Definition Files** (.OL-printerdef):
  `C:\Users\[UserName]\Connect\workspace\configurations\PrinterDefinitionConfig`
- **OMR Marks Configuration Files** (.hcf):
  `C:\Users\[UserName]\Connect\workspace\configurations\HCFFiles`

Where `[username]` is replaced by the appropriate Windows user name.

Tip

Actually, the path may not begin with 'C:\Users', as this is language-dependent. On a French system, for example, it would be 'C:\Utilisateurs'.

Type `%userprofile%` in a Windows File Explorer and press Enter to open the actual current user's home directory.

Other Resources

- **OL Connect Designer Templates** or **Package** files, copied from the folder where they reside.
- All PostScript, TrueType, Open Type and other **host based fonts** used in templates must be reinstalled on the new workstation.
- Import all **dynamic images** and make sure their paths match those in the old server.
- Make sure the new workstation can also access network or remote images, JavaScript, CSS, JSON, and HTML resources referenced in the Connect templates.

Secondary software and licenses

The following only applies to specific secondary products and licenses that interact or are integrated into the main product.

Image, Fax and Search Modules

- Reconfigure the Image and Fax outputs with the new host information.
- Import the Search Profile and rebuild the database in order to generate the database structure required by the Workflow.

Capture

1. Download the latest version of the Anoto PenDirector.
2. Before installing the PenDirector, make sure the pen’s docking station isn’t plugged into the server. Then install the PenDirector.
3. Stop the Messenger 8 service on the old and new server from the Workflow menu bar: **Tools > Service Console > Messenger** > right-click and select **Stop**.

4. Import the following files and folders from the old server into their equivalent location on the new server:
   `C:\ProgramData\Objectif Lune\PlanetPress Workflow 8\PlanetPress Watch\capture\PPCaptureDefault.mdb`
   `C:\ProgramData\Objectif Lune\PlanetPress Workflow 8\PlanetPress Watch\DocumentManager`
   `C:\ProgramData\Objectif Lune\PlanetPress Workflow 8\PlanetPress Watch\PGC`
5. If Capture was previously using an external MariaDB, MySQL or Microsoft SQL Server, reconfigure the ODBC connection details as previously from the Workflow Preferences by clicking on the Workflow button at the top left corner and clicking on Preferences, then reconfigure the PlanetPress Capture options under **Behavior >PlanetPress Capture > Use ODBC Database**.
6. Start the Messenger 8 service on new server from the Workflow menu bar: **Tools > Service Console > Messenger** > right-click and select **Start**.

OL Connect Send

- As of version 8.6 the Connect Send plugins are installed automatically with Workflow. If you are using an older version, run the OL Connect Send Plug-in Installer on the new Workstation to re-install the Connect Send plugins.
- Reconfigure the Server URL and port during the OL Connect Send Printer Driver setup.
- Re-run the OL Connect Send printer driver setup on client system and select the Repair option to point the clients to the new Server URL.

Configuring the Connect Engines

Any changes made to the Server preferences require the OLConnect_Server service to be restarted to take effect.

1. Stop the OLConnect_Server service from **Control Panel > Administrative Tools > Services > OLConnect_Server > Stop**.
2. Configure the Merge and Weaver Engines scheduling preferences as in the previous installation
   - Open the Server Configuration from:
     `C:\Program Files\Objectif Lune\OL Connect\Connect Server Configuration\ServerConfig.exe`

- Configure the DataMapper, Merge and Weaver engines preferences (see "Parallel Processing preferences" on page 88). As of version 2018.1 these preferences include the minimum (Xms) and maximum (Xmx) memory utilization for the Merge and Weaver engines.
- Configure any other options for the Clean-up Service.

3. Now start the **OLConnect_Server** service

Configuring the Server Extensions

In the case where the OLConnect MariaDB is installed on the new Master Server, it is important to reconnect all Server Extension systems to the new Master Server.

Perform the following action on each Server Extension:

1. Stop the OLConnect_ServerExtension service from **Control Panel > Administrative Tools > Services > OLConnect_ServerExtension > Stop**.
2. Open the Server Extension Configuration from:
   `C:\Program Files\Objectif Lune\OL Connect\Connect Server Extension\ServerExtension.exe`
3. Click on Database Connection and configure the JDBC Database connection settings so that the hostname points to the new Master Server.
4. Click on Scheduling and type in the location of the new Master Server.
5. Start the **OLConnect_ServerExtension** service.

Transferring software licenses

Once all the above resources have been transferred over to the new server, it is recommended to thoroughly test the new system - in **demo mode** - with sample files under normal production load to identify points of improvement and make sure the output matches the user's expectation.
Output generated at this point will normally bear a watermark which can be removed by transferring licenses from the old server to the new one.

- To transfer Connect and Workflow licenses, the user is usually required to complete a **License Transfer Agreement** which can be obtained from their local Customer Care department.

- If you want to transfer your licenses to the new machine right away, you may ask your local Customer Care department for a **30day Transition activation** code for your *old* machine.
- Upgrades cannot be activated using the automated Activation Manager. Contact your local Customer Care department.

To apply the license file received from the Activation Team:

1. Ensure that all services are stopped on your old machine before activating and starting the services on the new machine. Attempting to run the software with the same license simultaneously will not only run into errors but it is a breach of our EULA.
2. Start the PReS Connect, PlanetPress Connect or PrintShopMail Connect Software Activation module:
   `C:\Program Files\Objectif Lune\OL Connect\Connect Software Activation\SoftwareActivation.exe`
3. Click on Load License File to import the license.OLConnectLicense.
4. Start the Software Activation module on the Extension servers, where applicable.
5. Click on Load License File to import the above same license.OLConnectLicense.
6. Restart the OLConnect_Server service and restart the OLConnectServer_Extension service on the Extension servers, where applicable.
7. The number of Expected Remote Merge and Weaver engines should now be configurable in the Connect Server Configuration module (`C:\Program Files\Objectif Lune\OL Connect\Connect Server Configuration\ServerConfig.exe`)

To apply the PlanetPress Capture License:

1. Open the Workflow Configuration.
2. Click on Help on the Menu Bar and click on PlanetPress Capture License manager to import your license.

Uninstalling PrintShop Mail Connect from the previous workstation

It is recommended to keep the previous install for a few days until everything is completed. However, once your transition is successful and complete, the OL Connect software must be uninstalled from the original server. See "Uninstalling" on page 100.

Upgrading

This page provides information about Upgrading to PrintShop Mail Connect version 2022.1.

Upgrade information is detailed in the following pages:

- "Upgrading from previous Connect versions" below

Upgrading from previous Connect versions

Always backup before upgrading

It is recommended that you always backup your existing Connect preferences before upgrading to a new version. This will enable you to revert back to the previous version, in a worst case scenario in which the new version introduces issues with your existing production processes. Whilst the probability of such a worst case scenario is remote, it cannot hurt to take some simple precautions, just in case.

For instructions on how to do so, please see "Backup existing Connect version" on the next page.

> **Note**
>
> The scheduling settings were changed significantly in version 2019.2. Please make sure to record your current scheduling settings for reference before proceeding with an upgrade.

Users of Connect prior to 2022.1

Users of Connect 2022.1 should see the page "Upgrading from Connect 2021.2 or earlier with Remove User Data CHECKED:" on page 57 for information about exactly what data is saved or deleted.

Users of Connect prior to 2019.1

Users of all other Connect versions prior to 2019.1 should note that **Update Client 1.2.40** is a prerequisite for both OL Connect 2019.1 and Connect Workflow 2019.1 installations. Only Update Client 1.2.40 has the capacity to upgrade the OL Connect license to the newer format that is required by the installers of those products.

If you do not have Update Client version 1.2.40 installed already, then the next time you run your Update Client it will show that there is an update available of itself to Version 1.2.40 (or later).

Simply click on the “Install” icon to initiate the upgrade.

For further details on how to upgrade the Update Client and update your Connect License see the Update Client 1.2.40 Upgrade Guide (http://help.objectiflune.com/en/archive/Update_Client_1.2.40-Upgrade_Guide.pdf).

Note that an erroneous or incomplete uninstall before a reinstall or upgrade to a newer version of OL Connect may cause issues (see Product or engine exits within a second of starting in Connect's Knowledge Base).

> **Note**
>
> If an error occurs during uninstallation or after/when re-installing Connect after uninstalling it, please see: Problems during a Connect installation or version upgrade in Connect's Knowledge Base (https://help.objectiflune.com/en/kb-connect/#KB/FAQ/General/Installation_Upgrade.htm).

Backup existing Connect version

It is recommended that you always backup your existing Connect preferences before upgrading to a new version. This will enable you to revert back to the previous version, in a worst case scenario in which the new version introduces issues with your existing production processes. Whilst the probability of such a worst case scenario is remote, it cannot hurt to take some simple precautions, just in case.

Backing up a virtual machine

Backing up a virtual machine installation is relatively straight forward. Simply take a snapshot of the virtual machine instance, prior to upgrading. This would save all the localized preferences and configurations.

Backing up a real machine

Note

The scheduling settings in version 2019.2 have totally changed, making it impossible to revert these settings back to the previous version. Please make sure to record your current Scheduling settings for reference before proceeding with an upgrade.

Backup these folders

- `C:\ProgramData\Objectif Lune\OL Connect\.settings\ConnectHostScope`
- `C:\Users\[UserName]\Connect\filestore`
- `C:\Users\[UserName]\Connect\workspace\configurations`
- `C:\Users\[UserName]\Connect\workspace\Designer\.metadata\.plugins\org.eclipse.core.runtime\.settings`
- `C:\Users\[UserName]\Connect\workspace\Server\.metadata\.plugins\org.eclipse.core.runtime\.settings`

Where `[username]` is replaced by the appropriate Windows user name.

Tip

Actually, the path may not begin with 'C:\Users', as this is language-dependent. On a French system, for example, it would be 'C:\Utilisateurs'.

Type `%userprofile%` in a Windows File Explorer and press Enter to open the actual current user's home directory.

Backup your database

If you want to be completely thorough and be able to exactly replicate your existing system, you should also backup your existing Connect database.

If the default (pre Connect 2022.1) MySQL database were being used as the Connect back-end database, we would recommend the MySQLDump tool be used for this. See for details on this utility program: mysqldump (https://dev.mysql.com/doc/refman/5.7/en/mysqldump.html).

Pre-existing User Data

The following scenarios display what happens to pre-existing User Data in a Connect upgrade.

> **Note**
>
> In regards to the MySQL (Connect version 2021.2 and earlier) or MariaDB (Connect version 2022.1 onwards) entries: these are only applicable if the OL Connect database component was installed in a previous Connect installation.

Scenarios

Upgrading from Connect 2021.2 or earlier with **Remove User Data** CHECKED:

- In all cases:
  - Files and folders are removed from the user data folder *C:\Users\<connectUser>\Connect* , where *<connectUser>* is the user that installed OL Connect.
  - Files and folders are removed from the following data folders:
    - *C:\ProgramData\Objectif Lune\OL Connect\.settings*
    - *C:\ProgramData\Objectif Lune\OL Connect\CloudLicense*
    - *C:\ProgramData\Objectif Lune\OL Connect\ErrorLogs*
    - *C:\ProgramData\Objectif Lune\OL Connect\LiquibaseUpdate*
  - Files are removed from the root of the data folder *C:\ProgramData\Objectif Lune\OL Connect*.
    If the folder is empty following this (i.e. no license or user folders were present) then the *C:\ProgramData\Objectif Lune\OL Connect* folder itself is removed.
  - License files as well as any content not listed above but found in the *C:\ProgramData\Objectif Lune\OL Connect* folder remain untouched.
- Additional cases:
  1. If MySQL was previously installed as an OL Connect component AND the database contains some user-defined schemas:

        - The native OL Connect schema is removed from the MySQL database.
        - The MySQL database files (*C:\ProgramData\Objectif Lune\OL Connect\MySQL*) are kept intact, as user-defined schemas mean that the user did not have only the OL Connect native schema content in their database.
            - A message at the end of the upgrade will advise the user that some non-OL schemas were found in the database, so the database files were not removed.
    2. If MySQL was previously installed as an OL Connect component AND the database does not contain any user-defined schemas:
        - The MySQL database files (*C:\ProgramData\Objectif Lune\OL Connect\MySQL*) are removed entirely.

Upgrading from Connect 2021.2 or earlier with **Remove User Data** UNCHECKED:

- In all cases:
    - The user data folder *C:\Users\<connectUser>\Connect* (where *<connectUser>* is the user that installed OL Connect) is retained, untouched.
    - All the files and folders under the data folder *C:\ProgramData\Objectif Lune\OL Connect* remain untouched.
- If MySQL was previously installed as an OL Connect component:
    - All schemas from the MySQL database are migrated to MariaDB, allowing the user to continue using their database content normally.
      NOTE: This might take some time during the installation, depending upon the size of the existing databases.
    - The MySQL database files (*C:\ProgramData\Objectif Lune\OL Connect\MySQL*) are also kept intact.

Uninstalling Connect 2022.1 or above with **Remove User Data** CHECKED:

- In all cases:
    - Files and folders are removed from the user data folder *C:\Users\<connectUser>\Connect* , where *<connectUser>* is the user that installed OL Connect.

- Files and folders are removed from the following data folders:
  - *C:\ProgramData\Objectif Lune\OL Connect\.settings*
  - *C:\ProgramData\Objectif Lune\OL Connect\CloudLicense*
  - *C:\ProgramData\Objectif Lune\OL Connect\ErrorLogs*
  - *C:\ProgramData\Objectif Lune\OL Connect\LiquibaseUpdate*
- Files are removed from the root of the data folder *C:\ProgramData\Objectif Lune\OL Connect*.
  If the folder is empty following this (i.e. no license or user folders were present) then the *C:\ProgramData\Objectif Lune\OL Connect* folder itself is removed.
- License files as well as any content not listed above but found in the *C:\ProgramData\Objectif Lune\OL Connect* folder remain untouched.

- Additional cases:
  1. If MariaDB was previously installed as an OL Connect component AND the database contains some user-defined schemas:
     - The native OL Connect schema is removed from the MariaDB database.
     - The MariaDB database files (*C:\ProgramData\Objectif Lune\OL Connect\MariaDB*) are kept intact, as user-defined schemas mean that the user did not have only the OL Connect native schema content in their database.
       - A message at the end of the upgrade will be displayed, stating some non-OL schemas were found in the database, so the database files were not removed.
  2. If MariaDB was previously installed as an OL Connect component AND the database does not contain some user-defined schemas:
     - The MariaDB database files (*C:\ProgramData\Objectif Lune\OL Connect\MariaDB*) are removed entirely.

Uninstalling Connect 2022.1 or above with **Remove User Data** UNCHECKED:

- In all cases:
  - The user data folder *C:\Users\<connectUser>\Connect* (where *<connectUser>* is the user that installed OL Connect) is retained, untouched.

    - All the files and folders under the data folder *C:\ProgramData\Objectif Lune\OL Connect* remain untouched.

- If MariaDB was previously installed as an OL Connect component:
    - All schemas from the MariaDB database are kept, allowing the user to use those database files if they reinstall the software.

Preferences

The Preferences dialog is used to modify the general software preferences. Changes made in this dialog affect the software globally, not individual templates and data mapping configurations.

The Preferences dialog is separated into individual tabs, where each tab controls certain aspects of the software.

To open the Preferences dialog, select **Window > Preferences**.

The General preferences also provides you with buttons to :

- **Reset all Warning Dialogs**: This re-enables all warning dialogs that might have been previously disabled by selecting the "Don't show again" checkbox within the dialog.
- **Restore Defaults**. This option restores the preferences to Defaults. This applies to the current Preferences page only, but not other Preferences.
- **Apply**: This option applies the settings made within the current Preferences page, but does not close the Preferences dialog.

Clean-up Service preferences

The Clean-up Service defines how the Connect database and the temporary files created during Connect production runs are cleaned up after the production run has finished.

As part of the job production process PrintShop Mail Connect uses a database for intermediate storage and also creates various temporary "managed" files. These files include data extractions, configuration files and any intermediate files created during the production process. Connect keeps track of all these files through references held within the Connect database.

All the files created and the database references to them are stored for a set amount of time in order to allow Connect to reuse them. However, we do not want to store these indefinitely, because the database would run out of space. The solution is to use the "Clean-up Service" to remove the temporary data and files once they are no longer needed.

The more items that are present in the database, and the larger they are, the more time and processing power (CPU) that will be required for cleaning them up. Thus a regular Clean-up of the database (as often as possible) is recommended.

This is especially the case if items are not going to be retrieved from the database at a later date. i.e. If the Connect job is not going to be re-run.

The clean-up can always be set to run outside of business hours (see the "Run according to the cron schedule: Enter the interval at which the Clean-up service runs. To understand how to write a cron job schedule, please refer to the Quartz Scheduler tutorial: http://www.quartz-scheduler.org/documentation/quartz-2.3.0/tutorials/tutorial-lesson-06.html." below option below), to reduce impact upon Production systems.

The values below define when the specified targets are to be *set* as being ready for deletion, not *when* they are actually deleted. The actual deletion occurs only as per the cron job scheduling; or when PrintShop Mail Connect is started (if **Run at application start up** is selected); or when the **Run Now** button is pressed.

- **Enable clean-up service**: Check to enable the Clean-up services. When checked, either or both of the *Database clean-up* and *File clean-up* services can be set individually.
  If the box is not checked, then no Clean-up will occur.
- **Run at application start up**: Click to start the clean-up service when the Designer module is opened.
- **Run according to the cron schedule**: Enter the interval at which the Clean-up service runs.
  To understand how to write a cron job schedule, please refer to the Quartz Scheduler tutorial: http://www.quartz-scheduler.org/documentation/quartz-2.3.0/tutorials/tutorial-lesson-06.html.

Note

If the **Product managing the service** is set to Designer, then the Designer *must be running* at the time that the cron job is scheduled, for the Clean-up to run.

- **Product managing the service**: Select which of the applications will run the service.For PrintShop Mail Connect the Designer is the only choice.
- **Database Clean-up Service**:
  - **Allow database clean-up service**: Select this checkbox to enable the database Clean-up settings, and enable the actual clean-up.
    - **Threads to use for database deletions**: The number of Threads to be used in the clean-up. PrintShop Mail Connect is a multi-threaded application, and the clean-up is likewise.

      Tip

      The default number of threads is considered the best compromise for running both clean-up and production jobs simultaneously. If experience suggests that the clean-up is not running efficiently, then upping the number of threads here would be recommended. Conversely, if production appears to be suffering courtesy of the clean-up process, then reduce the number of threads here.

      In general, higher end machines (those with multiple cores) will allow a higher numbers of threads, whilst low end machines will perform better with a lower number of threads.

    - **Number of entities in each deletion batch**: The number of entities to be deleted at a time. This is done to break the clean-up into smaller chunks. This improves PrintShop Mail Connect clean-up responsiveness, whilst the clean-up is occurring.
      The number selected here applies to all the following settings.
      i.e. a selection of 1,000 would delete 1,000 data records within a **Data Set**, 1,000 content items within a **Content Set**, and so on.

- **Minimum time to retain Data Sets**: The minimum time a Data Set (and all the records it contains) is retained within the database before being set for deletion.

  > **Tip**
  >
  > In order to prevent attempts at deleting database objects which might still be in use, it is recommended that all **Minimum time** retention values *should always be set to at least the length of your longest job*. Preferably with some extra time added, for good measure.
  >
  > For example, if your longest job takes 45 to 50 minutes to run, then set the retention time to 55 minutes (or 1 hour) to ensure that all the database objects created during job processing survive for the duration of the job.

- **Minimum time to retain Content Sets**: The minimum time a Content Set (and all the content items it contains) is retained within the database before being set for deletion.
- **Minimum time to retain Job Sets**: The minimum time a Job Set (and all the jobs information it contains) is retained within the database before being set for deletion.
- **Minimum time to retain Managed Files**: The minimum time file references (to files such as data mapping configurations and templates) are retained within the database before being set for deletion.
- **Minimum time to retain other entities**: The minimum time any orphaned data (such as Finishing tables, Media tables, DataModels and Properties tables) are retained within the database before being set for deletion.

- **Database Partition Settings**:
  - **Use Database Partitioning**: Select to use Database Partitioning.
    - **Empty partition count**: The number of empty partitions that are created each clean-up run. This defaults to 24.
    - **Partition Size**: Enter the length of time before partitions are switched. This can be entered in minutes, hours, days, weeks or months.
- **File Clean-up Service**:

- **Allow file clean-up service**: Check to automatically detect orphan files and set them for deletion. Orphan files could be resources and internal files used by Connect, but which are not needed by any running job.
    - **Minimum time to retain orphaned files**: The minimum time during which orphaned files are kept in the database before being set for deletion.

The Clean-up Services preferences also provides you with buttons to :

- **Run Now**: This will run the clean-up service immediately.
- **Restore Defaults**. This option restores the preferences to Defaults. This applies to the current Preferences page only, but not other Preferences.
- **Apply**: This option applies the settings made within the current Preferences page, but does not close the Preferences dialog.

Note that this setting only impacts how dates are displayed in the Data Model. It does not change how dates are stored in the Connect database, nor how they are displayed in a template.

Database Connection preferences

Dialog used to change the PrintShop Mail Connect back-end Database.
This dialog supports the swapping of the back-end database between various vendor databases. Note, however, that the alternate vendor database(s) *must already be installed and available* in order to swap to them.

This is not a migration tool. It is a simple connection tool, that enables shifting to a different back-end database. Any existing data will **not** be transferred/ migrated between the databases, and any existing File Stores will be cleansed by the Clean-up Service after the swap.

When shifting to a different back-end database, the changes won't be applied until PrintShop Mail Connect is restarted. Including the Connect services. A full machine restart is recommended, as this provides the cleanest restart of all the services.

- **Basic Connection Settings** selections:
  - **Database vendor**: Select the database type from the drop down list.

    > **Note**
    >
    > Moving from one vendor database to another will reset all screen selections to defaults, regardless of what may have been previously selected.

  - **Database URL**: This is a read-only summation of the current database connection settings.

    > **Tip**
    >
    > If the **Test Connection** button shows that the database cannot be successfully connected to using the selected settings, then the contents of this field could be used to try to connect to the database outside of PrintShop Mail Connect. This should help in determining and refining the acceptable connection options.

  - **Hostname**: Enter the IP Address or alias of the server where database resides.
  - **Database Instance Name**: Enter an existing Microsoft SQL Server's instance name.
    This option only applies to existing Microsoft SQL Server instances, and not for MariaDB or MySQL.
  - **Port**: Enter Port number. The defaults are those which the vendors use by default.
  - **Schema**: The individual database schema, within the vendor database.

    > **Note**
    >
    > If a previously non-existent schema were chosen here, then a new schema of that name will be created within the database when the back-end database swap is applied. The tables within that schema, though, will not be created

until Connect is restarted.

- **Username**: Enter the database login username.

> **Tip**
>
> It is considered best practice for this user to have root privileges.

- **Password**: Enter the password associated with selected username.

- **Advanced Connection Settings** selections:
  - **Maximum concurrent threads**: This option sets the maximum database threads. The maximum setting is determined by the specific capabilities of the machine Connect is installed upon (CPU speed and the amount of cores being the major determinants).

> **Tip**
>
> Leaving this value set to the default maximum *should* be the best option in most circumstances.
> We recommended this entry be left at the default value.

  - **Custom database parameters** table: These are extra parameters which are appended to the database connection URL. The default values are those which have been determined to be useful in connecting to specific vendor databases. For example, in order to use Integrated Authentication with Windows, specify the property `integratedAuthentication` with the value `true`, which adds `integratedAuthentication=true` to the URL.

- **Property**: These are free field text fields.

  > **Note**
  >
  > These fields and their associated values get appended to the JDBC connection and therefore must follow all rules regarding acceptable URL addresses for such.

  - **Value**: The value applied to the associated Property.
  - **Add**: Used to add extra Property values to the table.
  - **Delete**: Used to remove existing Property values from the table.
- **Test Connection**: Use to test if current connection settings will connect to the specified database.
- **Restore Defaults**: Will restore the settings to PrintShop Mail Connect HyperSQL standard defaults.
- **Apply**: When a database connection is confirmed as correct this button becomes active, and is used to actually apply the database swap.

The Database Connection preferences also provides you with buttons to :

- **Test Connection**: This will run a test on the current Database Connection settings.
- **Restore Defaults**. This option restores the preferences to Defaults. This applies to the current Preferences page only, but not other Preferences.
- **Apply**: This option applies the settings made within the current Preferences page, but does not close the Preferences dialog.

Editing preferences

These preferences define different editing options in the Designer module.

- **Object Resizing for elements**: This defines in which contexts to enable the resizing of elements (including Positioned and Inline boxes). Resizing elements may cause layouts to produce undesirable results especially when using Foundation templates.

    - **Enable for Print Context**: Check to enable resizing in the Print contexts.
    - **Enable for Email Context**: Check to enable resizing in the Email contexts.
- **Detail tables preview limit**
    - **Maximum number of records to show in preview**: This setting limits the number of records to show in detail tables in the Designer (Preview tab). The lower this number, the shorter the time before the preview or the next record will appear.
- **Dragging data fields to the editor**. When you drag-and-drop a data field into a template, a placeholder appears in the template and a script is added (see "Inserting variable data directly (drag-and-drop)" on page 351). This option specifies what to use as the script's selector.
    - **New scripts use a class selector (recommended)**: The placeholder is wrapped in a Span element, and the Span's class is used as selector. This is recommended because it is faster (see: "Optimizing scripts" on page 408).
    - **New scripts perform a search and replace**: The placeholder text is used as selector.

> **Tip**
>
> Hold the **ALT** key while dragging to change this behavior on the fly.

CSS options

A CSS preprocessor is a CSS extension language that allows you to enhance CSS with code (variables, for example) and then compile it into plain CSS. CSS Preprocessor **Sass** is integrated in Connect.
For more information about Sass, see: https://sass-lang.com/.

The CSS options relate to the way Sass (.scss) files are compiled in Connect. See also: "Using a Sass file" on page 307 in the Online Help: https://help.objectiflune.com/en/planetpress-connect-user-guide/2022.1/.

- **Output style**: This setting determines the use of spaces and indentation in the compiled CSS file. For examples of the available output styles, see: https://sass-lang.com/documentation/file.SASS_REFERENCE.html#output_style.
    - **Compact**: Each CSS rule takes up only one line, with every property defined on that line. Nested rules are placed next to each other with no new line, while

separate groups of rules have new lines between them.

- **Compressed**: This output style minifies the output. It has no whitespace except that necessary to separate selectors and a new line at the end of the file. It also includes some other minor compressions, such as choosing the smallest representation for colors. It's not meant to be human-readable.
- **Expanded**: This is the default output style. Each property and rule take up one line. Properties are indented within the rules, but the rules aren't indented in any special way.
- **Nested**: Each property has its own line, but the indentation isn't constant. Each rule is indented based on how deeply an element is nested in the HTML and CSS structure.

- **Auto compile on saving .scss files**: When this option is checked, a .scss file is compiled into a .css file whenever you save it, overwriting any previously compiled version of the .css file. Also, when a *partial* .scss file is saved all .scss files are recompiled. (A partial .scss file is meant to be imported in another .scss file. Its name starts with an underscore.)
  By default this option is enabled.

Note

- Re-compiling a .scss file overwrites any manual changes made to the .css file.
- Partial .scss files cannot be compiled.
- Single line comments (//...) are not added to the compiled .css file, whereas multi-line comments (/* ... */) are maintained. Multi-line comments should be added within the brackets of the definition they apply to, to make sure they appear in the correct place in the compiled css.

Color options

Many of the colors in the user interface of Connect Designer can be adjusted. Click the small colored square next to the field that holds the default color value, to open the Color dialog and pick a color (see "Color Picker" on page 492 in the Online Help: https://help.objectiflune.com/en/planetpress-connect-user-guide/2022.1/).

- **Edges**: The edges around elements in a section in the Workspace. Click the Show Edges icon to toggle the visibility of these edges.

    - **Box Objects**: This color highlights positioned boxes, inline boxes and Div elements; see "Boxes" on page 256.
    - **Table**: This color highlights tables, and the rows and columns in tables; see "Table" on page 278.
    - **Resizable Table**: This color highlights tables for which the option Allow resizing has been checked when adding the table; see "Table" on page 278.
    - **Forms**: This color highlights forms; see Forms.
    - **Shared Content**: This color highlights shared content, such as shared snippets; see "Snippets" on page 285.
- **Margin and guides**: These settings only apply to Print sections.
    - **Guides**: This is the color for rulers that can help position content correctly; see "Guides" on page 314.
    - **Margins**: This color delineates the content area on a page; see "Pages" on page 147.
    - **Bleed box**: This color delineates the printable area on a page; see "Page settings: size, margins and bleed" on page 148.
- **Master pages**: These edges are only visible on Master pages; see "Master Pages" on page 156.
    - **Header and Footer Margin**: This color highlights the header and footer margin set for the Master page; see "Adding a header and footer" on page 157.
    - **Objects**: This color highlights all elements on the Master page.
- **Script Result Highlighter**:
    - **Results**: Hovering over a script in the Scripts pane highlights content that will be affected by the script; see "Personalizing content" on page 338.

Images preferences

- **Transparent PDF image preview**: Check this option so that PDF resources added to the template (including in the Master Page and Media) display using transparency. Note that this can affect display performance (showing transparent PDFs is slower) but will not affect output speed.

The Editing preferences also provides you with buttons to :

- **Restore Defaults**. This option restores the preferences to Defaults. This applies to the current Preferences page only, but not other Preferences.

- **Apply**: This option applies the settings made within the current Preferences page, but does not close the Preferences dialog.

Email preferences

Email (General) preferences

- **Default From Group**:
  - **Name**: Enter the name that is set by default in the "From name" field in the Send Email and Send Test Email dialogs ("Send (Test) Email" on page 543).
  - **Email Address**: Enter the email that is set by default in the "From Email" field in the Send Email and Send Test Email dialogs ("Send (Test) Email" on page 543).
- **Litmus account Group**:
  - **Email Test address**: If you have a Litmus account, enter the test address to use when sending a test email (see "Send (Test) Email" on page 543 in the Online Help: https://help.objectiflune.com/en/planetpress-connect-user-guide/2022.1/). For more information on Litmus, please see http://litmus.com/.

Email (SMTP) preferences

SMTP server presets can be selected when sending emails using either the Send Email or Send Test Email dialog. (See "Send (Test) Email" on page 543 and "Email header settings" on page 180 in the Online Help: https://help.objectiflune.com/en/planetpress-connect-user-guide/2022.1/). For all presets, the password is not saved and must be re-entered when sending emails.

- The **Add**, **Edit** and **Delete** buttons let you create and manage the presets.
- **SMTP Host Settings**: These settings can be made or edited after clicking the Add or Edit button.
  - **Name**: The name of the preset. This will show up in the Send Email dialog.
  - **Host**: The SMTP server through which the emails are to be sent. Can be a host (mail.domain.com) or an IP address.

> **Tip**
>
> Gmail only allows Connect to be used as an SMTP client when "Access for less secure apps" is enabled in the Google account settings.

- **Port**: The specified port number will be added to the host name, for example: smtp.mandrillapp.com:465.

> **Note**
>
> If the mail server supports it, the connection will be encrypted without the need to send the server a STARTTLS instruction when port 465 is used.

- **Use authentication**: Check if a user name and password are needed to send emails through the host.
- **Send STARTTLS**: Enabled if authentication is checked. With STARTTLS the client negotiates with the mail server to use some form of encryption, usually a version of Transport Layer Security (TLS). Since this improves security it is recommend to enable this option if you use port 25 (the default port), 2525, or 587.
  Note that the email will not be sent if the SMTP server does not support TLS or SSL (an older encryption type).
  This option is ignored when port 465 is used.
- **User**: Enter the user name used to connect to the SMTP server.

- **Restore Defaults**: There are three default presets, each for working with a different Email Service Provider (ESP): Mandrilapp.com, Sendgrid and Mailgun (see "Using an ESP with PrintShop Mail Connect" on page 1021).
- **Apply**: Apply the new settings without closing the Preferences dialog.

The Email preferences also provides you with buttons to :

- **Restore Defaults**. This option restores the preferences to Defaults. This applies to the current Preferences page only, but not other Preferences.
- **Apply**: This option applies the settings made within the current Preferences page, but does not close the Preferences dialog.

Emmet preferences

Emmet is a framework that enables the lightning-fast creation of HTML code though the use of a simple and effective shortcut language resembling CSS Selectors (see Emmet Abbreviations: "Emmet" on page 169. The Emmet functionality is available in the HTML and CSS source editors of Connect Designer. Emmet transforms abbreviations for HTML elements and CSS properties to the respective source code.

This is, for example, the abbreviation for a element with the class row:

```
div.row
```

On pressing the Tab key, this abbreviation is transformed to:

```
<div class="row"></div>
```

To learn more about Emmet itself, please see their website Emmet.io and their documentation: http://docs.emmet.io/. Emmet is a plugin. All options listed below are Emmet's default options. They are not specifically adjusted for Connect.

Common Emmet preferences

- **Expand abbreviations by Tab key**: Check to enable the Expand Abbreviation function.
- **... in files with extension**: Enter a comma-separated list of all file extensions in which expand abbreviation will work.
- **Upgrade web editors**: This Emmet option doesn't affect how Emmet works in Connect Designer.
- **Extensions Path**: Choose a folder where to put json and js files to extend Emmet. This includes custom snippets, preferences and syntax profiles. For more information see http://docs.emmet.io/customization/.

Emmet Abbreviation preferences

This Preferences tab lets you add and manage custom abbreviations. All standard abbreviations can be found in Emmet's documentation: http://docs.emmet.io/abbreviations/syntax/.

If there is no need to transform the text while expanding it, create an Emmet snippet instead (see below).

- **New**: Add a new abbreviation.
  - **Name**: The name of the abbreviation is also its trigger.
  - **Context**: The context in which the abbreviation is enabled (HTML, CSS, etc.).
  - **Description**: A short description of the abbreviation .
  - **Pattern**: This defines what an abbreviation expands to. Since Emmet is mostly used for writing HTML/XML tags, abbreviation definition uses XML format to describe elements; see http://docs.emmet.io/abbreviations/types/.
  - **Automatically insert**: This standard option doesn't affect how Emmet works in Connect Designer.
- **Edit**: Edit the currently selected abbreviation.
- **Remove**: Remove the currently selected abbreviation.
- **Import**: Click to open a browse dialog to import an XML file containing exported abbreviations. The imported abbreviations are added to the current list.
- **Export**: Click to open a Save as dialog to export all the abbreviations in an XML file that can be shared and re-imported.
- **Preview box**: Shows what the selected abbreviation is expanded to.
- **Restore Defaults**: clear all custom abbreviations.
- To temporarily disable an abbreviation, uncheck the checkbox next to the name of the abbreviation in the list.

Emmet Output preferences

The Output Preferences dialog is used to control how the expanded (output) code behaves when expanding abbreviations and snippets. There are 6 different dialogs to control output and, while they all have identical options, they control different output types: CSS, HAML, HTML, XML, XSL and the "Default" one controlling the rest of the types.

These options are equivalent to Emmet's syntaxProfiles.json feature (http://docs.emmet.io/customization/syntax-profiles/).

Emmet Snippets preferences

Emmet Snippet are similar to abbreviations in that they are expanded when the Tab key is pressed, but they are just blocks of plain text. Anything in a snippet will be outputted “as is”, without any transformation.

- **New**: Click to create a new snippet.
  - **Name**: The name of the abbreviation is also its trigger.
  - **Context**: The context in which the snippet is enabled (HTML, CSS, etc.).
  - **Description**: A short description of the snippet.
  - **Pattern**: The pattern defines what a snippet expands to.
  - **Automatically insert**: This option doesn't affect how Emmet works in Connect Designer.
- **Edit**: Modify the currently selected snippet.
- **Remove**: Remove the currently selected snippet from the list.
- **Import**: Click to open a browse dialog to import an XML file containing exported snippets. The imported snippets are added to the current list.
- **Export**: Click to open a Save as dialog to export all the snippets in an XML file that can be shared and re-imported.
- **Preview box**: Shows what the selected snippet is expanded to.
- To temporarily disable a snippet, uncheck the checkbox next to the name of the snippet in the list.

Emmet Variables preferences

Variables are placeholders used in Emmet snippets to output predefined data. For example, the html:5 snippet of HTML syntax has the following definition:

```
<!doctype html>\n<html lang="${lang}">...</body>\n</html>
```

In the example above, ${lang} is used to refer lang variable defined in variables below. If your primary language is, for example, Russian, you can simply override lang variable with ru value and keep the original snippets. Also, you can override variable values with inline abbreviation attributes: html:5[lang=ru].

- **Name**: The name of the variable. This should be a single alphanumeric string with no spaces or special characters. For example, the myVar name is referred to as ${myVar}.
- **Value**: The value of the variable when the snippet is expanded.
- **New**: Click to create a new variable and define its name and value.
- **Edit**: Click to modify the currently selected Variable.
- **Remove**: Click to delete the currently selected Variable.

The Emmet preferences also provides you with buttons to :

- **Reload Engine**. This allows you to reload the Emmet engine based upon the current settings.
- **Restore Defaults**. This option restores the preferences to Defaults. This applies to the current Preferences page only, but not other Preferences.
- **Apply**: This option applies the settings made within the current Preferences page, but does not close the Preferences dialog.

Engines preferences

See Engines Preferences.

Hardware for Digital Signing preferences

Connect can access secure hardware devices (USB Tokens, Smart Cards, and Hardware Security Modules) through PKCS#11, for use in creating PDF digital signatures.

This Preferences page allows for the configuration of these secure hardware devices. It caters for the two conflicting needs of Connect users.

1. Connect user wanting to create digital signatures using a secure device have to configure the DLL for using the secure device, so Connect can find the DLL when a digital signature is necessary.
2. The Connect administrator wanting the configuration of DLLs loaded by Connect Server, Designer, and their engines to happen outside of Presets, Templates, or other resources. This so the administrator can control what DLLs are loaded instead of allowing user created configurations triggering third party DLLs at runtime.

This preferences page caters for both needs.

The **PKCS#11 Modules and Tokens** table contains a list of Hardware devices that can be used by Connect.
Hardware devices can be added, modified or deleted from the table using the following buttons:

- **Add**: Add a new hardware device to the list. This launches the Add PKCS#11 Module dialog.
- **Edit**: Edit that information relating to an existing hardware device. This launches the Edit PKCS#11 Module dialog.
- **Delete**: Remove an existing hardware device from the list.

The Hardware for Digital Signing preferences also provides you with buttons to :

- **Restore Defaults**. This option restores the preferences to Defaults. This applies to the current Preferences page only, but not other Preferences.
- **Apply**: This option applies the settings made within the current Preferences page, but does not close the Preferences dialog.

Language preferences

- **Display language**: Select a language from the drop-down list to be used as the language of the User Interface (after the software is restarted).
- **Default Locale**: The default locale sets the locale for new templates. By default this is the system's locale. The locale can be changed per template; see "Locale" on page 336.
  - Select System Locale to use the operating system's locale settings.
  - Select Explicit Locale to choose a static locale from the drop-down list.

Default Locale

The Default Locale preferences are only available in the Designer Preferences. This setting determines the locale for new templates. By default this is the system's locale. Select *System Locale* to use the operating system's locale settings. Select *Explicit Locale* to choose a specific locale from the drop-down **Explicit Locale** list. The Locale can be changed on a per template basis. See "Locale" on page 336.

The Language Settings and Default Locale preferences also provides you with buttons to :

- **Restore Defaults**. This option restores the preferences to Defaults. This applies to the current Preferences page only, but not other Preferences.
- **Apply**: This option applies the settings made within the current Preferences page, but does not close the Preferences dialog.

Logging preferences

PrintShop Mail Connect logs the activities it undertakes whilst running. New Connect logs are created daily and are held for a period before they are automatically deleted.

The settings on this page determine the level of logging and how long the log files should be

retained.

These log files can be an essential resource when diagnosing issues with OL Support. The logging settings are global to ***all*** Connect applications and the Logging Preferences can be adjusted from within any of the Designer and Print Manager dialogs.

The settings are:

- **Overall Logging Level** selection: Select the overall Connect Logging level. This controls how much logging Connect will do.
  By default the logging level is set at the midpoint of *Info*, but it can be set higher to include more logging (*All*, *Trace*, *Debug*), or lower to reduce the amount of logging (*Warning*, *Error*).
  The logging is hierarchical, with each Logging Level selection containing all of the Logging Levels below it. The default logging *Info* Level contains all *Error* and *Warning* entries plus informational log entries.

> **Warning**
>
> Higher logging settings will have an impact upon Connect production speeds, as well as leading to substantially larger log files.
>
> We recommended leaving the logging level to *Info* and only using the higher levels of logging in conjunction with advice from OL support.

- **Rollover policy** selection: Chose whether to retain Connect log files for a certain number of days (**Daily logs**) or based upon some predetermined hard disk usage limitations (**Size-based logs**). The selection are as follows:
  - **Daily Logs**: Use this setting to determine how many days Connect Logs are to be kept, before they are deleted.
    - **Number of days to retain logs**: This value only impacts upon historic (closed) logs.
      Chose between 1 and 99,999 days.
      The default value is set to 10 days for a new installation and 99,999 days on existing installations (to preserve backward compatibility).
  - **Size-based logs**: Use this setting to restrict log file size, and to keep only a specified number of them.
    By combining the maximize individual log file size with the amount of log files to

retain, this effectively allows a hard disk space usage limitation to be placed upon the logging process.

  - **Maximum size for log file**: This sets the maximum size a log file can reach before the logging system creates a new file.
  - **Number of files to keep**:This sets the maximum number of log files kept in the log folder. The default value is set to 50 for a new Connect installation and 99, 999 for an existing installation (to preserve backward compatibility).

- **Logging pattern** edit box: This edit box determines the formatting of the individual log entries.
  By default it is set to a date/time value which allows for simple searching within log files.

> **Warning**
>
> We recommend leaving the Logging pattern to the default value.
>
> If you do need to change the Logging pattern, please see the Pattern Formatting guide (https://logback.qos.ch/manual/layouts.html#conversionWord) for help in doing so.

- **Log message preview** display: This displays a real time example of the format and content of individual log file entries, based upon the **Logging pattern** setting.

> **Note**
>
> The **language** of the log files is determined by the Language preferences.

Advanced settings

The Advanced settings over-ride the Overall logging settings, and provide a greater level of logging granularity.

> **Warning**
>
> Higher logging settings will have an impact upon Connect production speeds, as well as

leading to substantially larger log files.

The Advanced Log Settings should only be set in conjunction with advice from OL support, to ensure that only the most relevant settings are set to the higher logging levels.

This Preferences page allows you to add ( ) or remove ( ) individual Connect Packages, or change their logging settings ( ).

The Log Setting and Advanced Log Setting preferences also provides you with buttons to :

- **Restore Defaults**. This option restores the preferences to Defaults. This applies to the current Preferences page only, but not other Preferences.
- **Apply**: This option applies the settings made within the current Preferences page, but does not close the Preferences dialog.

Parallel Processing preferences

See "Parallel Processing preferences" on page 88.

Print preferences

Available Printers preferences

The Available Printers preferences control which printer definitions are available when generating print output or creating Output Presets. Any printer that is unchecked in this dialog will not be visible in the "Model" drop-down of the Print Options dialog; see "Print options" on page 704 and "Adding print output Models to the Print Wizard" on page 1005.

Available Printer Preferences:

- **Selected Printers**: Lists the available Printer Definition Files in the system. Note that these are not installed Windows printers or printer queues, but PrintShop Mail Connect Printer Definition Files.
- **Printer checkbox**: This checkbox selects/deselects all printers in the list. Click to check all, click again to uncheck all.

General Print preferences

In order to print, the Designer connects to a Connect Server. To enable communication between the Designer and the Connect Server, settings have to be made under "Preferences" on page 60.

Print Measurements preferences

- **Units**: Use the drop-down to specify the default measurements system used for dimensions of the template and boxes. In addition it defines the coordinates/position of box elements.
  The default unit will be added automatically when geometry values are entered without a unit in the Attributes pane or in the Box Properties dialog.
- **Flip insert guide axis**: Check this option to flip the axis on which guides are inserted. Normally, dragging a guide from a horizontal ruler inserts a horizontal guide (see "Guides" on page 314). With this option checked, dragging a guide from a horizontal ruler inserts a vertical guide.

The Print preferences also provides you with buttons to :

- **Test Print Server URL**. This button is only available for the General Print Preferences. It tests the Print Server URL settings made within that Preferences page.
- **Restore Defaults**. This option restores the preferences to Defaults. This applies to the current Preferences page only, but not other Preferences.
- **Apply**: This option applies the settings made within the current Preferences page, but does not close the Preferences dialog.

Sample Projects preferences

Sample Project deployment settings

- **Workflow Configurations**
  - **Encoding**: By default, Sample Projects set the encoding of the Workflow configuration that they create to the encoding of the system on which the Designer runs (**system default**).
    If Workflow runs on another machine, and that system has a different active code page, Workflow may be unable to correctly interpret all the characters that are used in the configuration file (in path or file names, for example). To prevent this, you may select an encoding from the list.

Save preferences

The saving preferences are a way control if and how often PrintShop Mail Connect saves your work in the background, and if how many backup files it creates when you save the template or data mapping configuration. See also: "Saving a template" on page 110.

Auto Save

After a template or data mapping configuration has been saved for the first time, Connect Designer can auto save it with a regular interval.

- **Enable**: activate the Auto Save function.
- **Interval (minutes)**: enter a number of minutes, e.g. 3 to auto-save the template or data mapping configuration every 3 minutes.

Auto Backup

Connect Designer can automatically create a backup file when you **manually** save a template or data mapping configuration. The Auto Save function does **not** cause backup files to be created.

- **Enable**: activate the Auto Backup function.
- **Revisions to keep**: Enter the maximum number of backup files. When the maximum is reached, Auto Backup will overwrite the oldest file.
- **Destination**: Select the directory in which the backups should be stored.
  - **Original**: the directory in which the original file is stored.
  - **Other directory**: use the **Browse** button to select another directory.

Backup files have the same name as the original file with two underscores and a progressive number (without leading zeros) at the end: **originalname__1.OL-template**, **originalname__2.OL-template**, etc.

The Save preferences also provides you with buttons to :

- **Restore Defaults**. This option restores the preferences to Defaults. This applies to the current Preferences page only, but not other Preferences.

- **Apply**: This option applies the settings made within the current Preferences page, but does not close the Preferences dialog.

Scripting preferences

The Scripting preferences define different options related to scripting within PrintShop Mail Connect. See also: "Testing scripts" on page 403.

- **General**:
  - **Script timeout at design time (sec)**: In Preview mode or when running the Script Profiler (see the Profile Scripts dialog), a long running script is stopped after the amount of time set here. The default is 2 seconds, the minimum is 1 second.
  - **Expanded script quotes style**: When the Expand button in a Script Wizard is clicked, the expanded script will use either **double (")** or **single (')** quotation marks.

    > **Tip**
    >
    > Using single quotation marks in a script simplifies adding HTML fragments, which typically use double quotes.

- **Designer scripting profiling group**:
  - **Number of iterations**: Enter the number of times to run scripts when running the Profile Scripts dialog. The default is 1000. Accepted values are 1 to 1000000000. Yes, that's 1 billion - which would take a *long* time to run!

The Scripting preferences also provides you with buttons to :

- **Restore Defaults**. This option restores the preferences to Defaults. This applies to the current Preferences page only, but not other Preferences.
- **Apply**: This option applies the settings made within the current Preferences page, but does not close the Preferences dialog.

Engines preferences

Background

A **Merge engine** merges the template and the data to create Email output, or to create an intermediary file for Printed output. The intermediary file is in turn used by a **Weaver engine** to prepare the Print output.

This page appears in the Designer **Window > Preferences** dialog. It allows control over Merge and Weaver (output) engines.

Settings

Configure engine options

- **Tasks handled by**:

  Select how Connect jobs are to be processed. The choices are between:

  - **Engines** (separate processes): Select to have a number of separate engines (for Merging and Weaver, as appropriate) to process jobs.

    > **Note**
    >
    > This is the best choice for production environments.

  - **Connect Designer** (uses less memory): Select to have a single Connect Designer handle processing.

    > **Note**
    >
    > This option should only really ever be used for low memory, non-production (testing) environments.

Number of engines

This group is only available if **Engines** were selected as the processing architecture .

- **Merge Engines** selection: Select the desired amount of Merge Engines.

  A **Merge engine** merges the template and the data to create Email output, or to create an intermediary file for Printed output. The intermediary file is in turn used by a **Weaver engine** to prepare the Print output.

  Generally, launching a relatively high number of **Merge** engines results in better performance, as Merge engines are involved in the creation of output of all kinds (Print, Email and Web) and because content creation is relatively time-consuming.

  > **Note**
  >
  > The Connect installer automatically sets the number of Merge Engines based upon the machine CPUs/Processors (the number of cores) and available Memory (RAM). The default setting will likely be enough for most users.
  >
  > However, virtual machines can provide an exception to this. In some virtual environments the Connect installer cannot accurately determine the number of logical cores actually available to the virtual machine, and will thus use a lesser number than that which could actually be supported.

  You can confirm how many Merge engines are running on your machine by running Task Manager and locating the mergeengine.exe entries in your process list.
  Note that there will be an extra Merge and Weaver engine in your task manager if you are running Designer.

  The Merge engine is responsible for the plugins Create Print Content, Create Email Content and Create Web Content.

  License restrictions only apply to the Merge engine when creating Email or Web content. They do not apply for Create Print Content. This means you are allowed to start an infinite amount of Merge engines on any given Connect Sever to run print jobs, but you will be restricted to a set number of Merge engines for your Email and HTML jobs.

- **Weaver Engines** selection: This option is *read-only* in PrintShop Mail Connect Designer Preferences.
  Select the desired amount of Weaver (Output) Engines.

  A **Merge engine** merges the template and the data to create Email output, or to create an intermediary file for Printed output. The intermediary file is in turn used by a **Weaver engine** to prepare the Print output.

  Adding extra Weaver (Output) engine(s) might be useful when large Print jobs are to be run simultaneously with smaller Print jobs. However, too many running engines will waste precious RAM and CPU cycles to idle processes, whilst too few could create a bottleneck.

Memory

This group is only available if **Engines** were selected as the processing architecture.

Specify the maximum amount of random access memory (RAM) in megabytes that will be used per engine in order to make optimal use of the machine's memory.

The selections made here will apply to all Engines of that type.

> **Note**
>
> These settings only control the maximum size of the *Java heap* memory that an engine can use; the total amount of memory that will used by an engine is actually a bit higher.
>
> Also keep in mind that the Connect Server and the operating system itself will need memory to keep running.

By default, each engine is set to use up to a predetermined amount of RAM. To make optimum use of a machine's capabilities it might be useful to increase the amount of memory that the various engines can use.

Here are some simple scenarios where you might want to increase an Engine's memory usage:

- DataMapper engines may perform better with greater memory when running jobs containing a lot of data.
- For complex templates with a lot of pages per document, there is a chance that Merge engines will run better with more memory.
- The maximum memory usage of a Weaver engine can be relevant for jobs with heavy graphics; or for jobs that use Cut & Stack impositioning; or for jobs using particular variables that entail page buffering (see "Content variables" on page 1012).

The options available in the **Memory** group are:

- **Merge Engines (MB)** selection: Select the desired amount of Merge Engine memory.

  For complex templates with a lot of pages per document, there is a chance that Merge engines will run better with more memory.
- **Weaver Engine (MB)** selection:
  Select the desired amount of Weaver (Output) Engine memory.

  When running really large jobs, it often pays to increase Weaver memory allocation, even if only for the duration of the production job(s).

  The maximum memory usage of a Weaver engine can be relevant for jobs with heavy graphics; or for jobs that use Cut & Stack impositioning; or for jobs using particular variables that entail page buffering (see "Content variables" on page 1012).

Multi-threading settings

This group allows you to set Multi-threading rasterization options for form ripping when generating AFPDS or IPDS output.

- **Enable multi-threaded reading** check box: This setting controls how the engine reads its input, with multi-threading reading being the default.

> **Note**
>
> This setting is primarily for trouble shooting. We recommended always leaving multi-threading on, unless otherwise advised by Support.

- **Number of multi-threaded form rips** selection: Set the amount of individual threads to assign to form ripping.

  This setting is useful for when graphics content triggers rasterization when writing AFP or IPDS output. Typical examples of content triggering rasterization are: transparency; fill patterns; gradient fills; or text rotated at a non-right angle.

  Increasing the number of threads should speed up Output Creation under such circumstances.

  > **Note**
  >
  > How effective this setting is at speeding up the output will depend upon the number of cores available on the processing machine and how many other tasks the machine is running simultaneously.
  >
  > If only one Output Creation task is running, setting this value as equal to the number of (logical) cores will likely give the best performance.

Buttons

The **Engines** preferences also provides you with buttons to :

- **Restore Defaults**. This option restores the preferences to Defaults. This applies to the current Preferences page only, but not other Preferences.
- **Apply**: This option applies the settings made within the current Preferences page, but does not close the Preferences dialog.

Parallel Processing preferences

The parallel processing preferences (previously referred to as Scheduling preferences, prior to 2019.2) page provides the means to control precisely how the PrintShop Mail Connect Designer handles jobs that operate in parallel.

For additional information on how these preferences can enhance performance, see Engine configuration and "Performance Considerations" on page 23.

Parallel Processing properties

Preset selection

Only the **Custom** setting is applicable to the PrintShop Mail Connect Preferences, so this option is always selected and the field made read-only.

Content Creation Tab

The options are:

- **Total Merge engines configured** read only display: This is a read only entry that shows the total number of Merge engines available. To change this value, you must update the Merge Engines in the "Engines preferences" on page 84 page.
- **Multi tasking** group:
  When starting a new Content Creation task, the task will immediately commence if there is a Merge engine available. How many Merge engines to use is based on the number of records in the input data.

  Select from the following options:

  - **Additional engine every (records)** entry: This controls how many Merge engines are used for a Content Creation task. It means that for every additional 'x' records in the task, an additional Merge engine will be used.
    For example, with the default 100 record threshold, tasks with 1-100 records will be assigned 1 Merge engine, tasks with 101-200 get assigned 2 merge engines, tasks with 201-300 get assigned 3 merge engines, and so on.
    The default of 100 records was chosen purely because it is an easily multiplied number, not because it has been proven to have any significant value. It means that on an average system (i.e., less than 10 Merge engines) any decently sized task is allowed to use all Merge engines. It also assumes that using more than one Merge engine for less than 100 records will probably not make a big enough difference to throughput speed. Obviously, there are situations where these assumptions will not apply.

    > **Note**
    >
    > Currently, it's only the print and PDF content creation tasks that use multiple Merge engines.

Buttons

The Parallel Processing preferences also provides you with buttons to :

- **Restore Defaults**. This option restores the preferences to Defaults. This applies to the current Preferences page only, but not other Preferences.
- **Apply**: This option applies the settings made within the current Preferences page, but does not close the Preferences dialog.

Known Issues

This page lists important information about issues that apply to PrintShop Mail Connect 2022.1.

Issue with CRLF in Go Sub runtime parameters

When CRLF characters are used in runtime parameters of the **GoSub** plugin, this causes an error. The issue is fixed in OL Connect 2022.2.

Installer issues

The new 2022.1 installer has some minor issues that will be fixed in a subsequent release. The issues are:

- The 'Select Setup Language' dialog language and the default language selection use the local machine language, but the list of language names remains in English, due to a limitation in the 3rd party tool being used.
- After installation, the "*recent files*" list is cleared and the measurement units are reset from 'cm' to 'inch'.
- When updating from earlier 2022.1.x versions the bundled MariaDB connection settings are reset.

Dynamic tables causing issues

In some specific circumstances the dynamic tables feature does not behave correctly. The following scenarios should be avoided to prevent subsequent issues with the template:

- Using `page-break-before:always` on the top TR of a TBODY causes a problem and results in a pagination error
- It is possible to specify conflicting attributes on a TR element:
  - `page-break-before:always` or `page-break-after:always` combined with data-repeat
  - `page-break-after:always` combined with `page-break-before:avoid` on the next sibling TR (or similar combinations)

These issues will be addressed in a later PrintShop Mail Connect release.

DataMapper: Automatic Date/Time does not work with certain negative UTC time zone offsets

The new "*Automatic*" date parsing option in the DataMapper cannot parse dates with negative UTC time zone offsets of non-zero minutes.

For example: *2021-03-01T10:00-03:**00*** will work, but *2021-03-01T10:00-03:**30*** will not.

Positive time zones with non-zero minutes (such as *+10:30*) are unaffected.

The negative time zone offsets affected are: **UTC -02:30**; **-03:30**; **-04:30** and **-09:30**

This issue will be fixed in a later release. In the meantime, the workaround is to use “*Custom*” or “*ISO8601*” date parsing options in DataMapper.

CSS inlining colour values now converted to RGB

As of PrintShop Mail Connect 2021.2 when using the CSS inlining mode "*Apply CSS properties on elements*" for emails, all colour values are now converted to RGB, rather than to HEX.

Issues running Connect on Hyper-V 9.0

Some customers have reported difficulties running PrintShop Mail Connect on Hyper-V version 9.0. In some instances PrintShop Mail Connect cannot install and in others the PrintShop Mail Connect Server service sometime stops with a signature error.
To resolve these issue we recommend downgrading to Hyper-V version 8.0 where these issues are not reported.

Minor font changes

As of Connect 2021.1 we no longer round fonts to pixel size. This can lead to tiny differences in the output (of 1-2 pixels) when compared to earlier versions.

Changed Omit Master Page Back behaviour

In versions of Connect prior to 2020.2, if a page had no content except for a linked DataMapper background, then Connect would consider it an "empty" page when determining whether or not to "*Omit Master Page Back in case of an empty back page*" (available as an option in the sheet configuration of a section). This has now been fixed in Connect 2020.2, and such pages are no longer considered empty.

This could impact on the output from existing templates.

Print Wizard Preview showing some Landscape mode jobs in Portrait mode

Print jobs featuring an associated Connect "Output Creation Presets" on page 1001 which have "Imposition options" on page 774 set to Landscape will not **Preview** in Landscape, but rather in Portrait mode. The printout is unaffected, however.

Issues associating PDF files with Connect

Under certain circumstances, Connect Setups prior to 2019.2 would fail when attempting to add the "Enhance with Connect" association with PDF files. This would then cause the setup to appear to fail.

Whilst this issue has been fixed in the Connect 2019.2 installer, if a user had previously experienced the issue and temporarily worked around it to complete the installation, then the Connect installer will fail on upgrade or uninstallation.

To get around this, a manual uninstall is required, or a modification to registry entries.

The license update introduced in OL Connect 2019.1 does not cater for existing AFP input licenses

AFP Input is an add on option for OL Connect licenses. Unfortunately, the update to the 2019.1 version of the OL Connect license does not cater for existing AFP input licenses.

If you have an existing AFP input license we ask that you contact your local Customer Care team after the initial license update is complete and have them add the AFP input option back into your license. (See (https://www.objectiflune.com/WebActivationManager/CareInfo.aspx.)

The Update Client OL for Connect 2019.1 does not cater for standalone Connect Print Manager installations

The Connect Print Manager is a standalone Connect tool that is used in production sites to manage print jobs. It can be installed and run on standalone machines without a Connect license.

The Update Client for Connect 2019.1 looks for a Connect license as part of the update process, but it will not find one on standalone Connect Print Manager installations. As a result the Update Client will not recognize that there is a Connect update available to the Print Manager machine.

In order to update Connect Print Manager to version 2019.1 you will need to download the Connect 2019.1 installer outside of the Update Client. The Connect 2019.1 installer can be downloaded from the Web Activation Manager site. See https://www.objectiflune.com/webactivationmanager/. Or you could ask your local Administration for the installer, as it would likely already have been downloaded for installation by the document designers.

Page break changes in 2019.1

Improved page breaking in Connect 2019.1 might impact upon some existing templates.
It is recommended that you check page breaking in existing jobs, where page breaks at a specific location are a known criteria.

Minor differences in AFP and IPDS output introduced in 2019.1

Connect 2019.1 now defaults to "*Scale to Fit*" for both **IPDS** and **AFP** output.
Please note that this may have an impact when printing pre-existing Connect jobs (made in Connect versions predating 2019.1) on older printer models that do not support "*Scale to Fit*".

Issue with image placement in 2019.1 when using some customized AFP and IPDS Printer Definitions

An issue with image placement in Connect 2019.1 AFP and IPDS output was discovered just prior to the release. The issue is specific to AFP and IPDS 600 DPI Printer Definitions that are not set to "*Scale to Fit*".

As "*Scale to Fit*" is now switched on by default in all standard Connect 2019.1 AFP and IPDS Printer Definitions, none of those should encounter the issue.

This issue will be fixed in a subsequent patch release.

Issue after erroneous or incomplete update or re-installation

If one or more products (the Connect Designer, Connect Server, Software Activation, Print Manager, or Server Configuration Tool) or engines exit within a second of starting, this may be caused by a recent erroneous or incomplete uninstall before a reinstall or upgrade to a newer version of OL Connect. (See Product or engine exits within a second of starting in Connect's Knowledge Base: http://help.objectiflune.com/en/kb-connect/#KB/FAQ/OL Connect/KB2019.htm.)

This may be solved by deleting the `%UserProfile%\Connect\.eclipse` directory.
For guidance on a full manual uninstallation please see the Solution in: Problems during a Connect installation or version upgrade in Connect's Knowledge Base: http://help.objectiflune.com/en/kb-connect/#KB/FAQ/OL Connect/KB2002.htm.

Backend database might require periodic maintenance

Databases maintain a variety of statistics in order to optimize performance. When high levels of inserts and/or deletions occur, the statistical data keeping can struggle to keep up. Over a period of prolonged and intensive processing this can result in a degradation in performance, with the whole database slowing down as it struggles to clean itself up.

In Connect terms the effect can be felt as the Data Mapper and/or Job Creation progressively slowing down.

To cure this issue, it is recommended that you periodically run manual maintenance on the backend database.

If using **MySQL**, the following script should be run in a query window:

```
set @a=null,@c=null,@b=concat("show tables where",ifnull(concat("
`Tables_in_",database(),"` like '",@c,"' and"),''),"  (@a:=concat_ws
(',',@a,`Tables_in_",database(),"`))");

Prepare `bd` from @b;
EXECUTE `bd`;
DEALLOCATE PREPARE `bd`;

set @a:=concat('optimize table ',@a);
PREPARE `sql` FROM @a;
EXECUTE `sql`;
DEALLOCATE PREPARE `sql`;

set @a=null,@b=null,@c=null;
```

If using **Microsoft SQL Server** run the following command in a query window:

```
sp_updatestats
```

Windows 10 Search service impacting Connect

The Windows 10 Search service runs as a background task, indexing files and folders. It has been noted that this background task is sometimes preventing files being added to the Connect temporary files folder when large amounts of files are being output and copied.

If this is an issue for you, we suggest disabling Search Indexing on the *C:\Users\<username>\Connect* folder.

This issue will be fixed in a later release.

Job Creation Presets: External Sorting change introduced in 2018.2

Versions prior to 2018.2 did not correctly save the line end characters for external sort configurations in Job Creation Presets, which meant the job could not be externally sorted. This issue has been fixed in version 2018.2. However, Job Creation Presets created with an earlier version may still have the wrong line end character for external sorting. To fix this, open the Job Creation Preset in the new version, reset the line end setting in the sorting options and then save the preset.

Business Graphics: Backward Compatibility Issues introduced in 2018.1

As a consequence of changes in both the user interface and the underlying technology, Business Graphics made with a version prior to PrintShop Mail Connect 2018.1 may not display correctly when opened in version 2022.1.

The currently known backward compatibility issues are listed here:

All charts

- **Legend position**: The position of the legend is not converted. It defaults to 'left' in a converted chart.
- **NOTE:** Expanded custom chart scripts cannot be converted.

Pie charts

- **Default colors**: The default colors (used when no pie chart colors are specified) have changed.

Known Font issues

The following font(s) are known to have issues in PrintShop Mail Connect 2022.1:

- **Benton Sans CFF** font

Minor differences in PCL, AFP and IPDS output introduced in 2018.1

The browser component (Mozilla Gecko) used in the WYSIWYG editor of the Designer was updated for Connect 2018.1. This allows use of new CSS properties, such as flexbox.

However this update could lead to increased output file sizes for some PCL, AFP and IPDS jobs. This is generally not a cause for concern, however there might be some associated increase in processing times, as well as some minor differences in the output. For example, table line widths and font spacings might differ slightly (particularly for SMALL CAPS text), which could lead to slightly different word-wrapping in some circumstances.

Windows Server 2016 issue

As of PrintShop Mail Connect 2018.1 Connect is officially supported under Windows Server 2016.

Please note, however, that the Upland Objectif Lune **Update Client** application might be blocked by the enhanced security settings in Windows Server 2016.

To fix this, add http://updates.ca.objectiflune.com to the list of trusted web sites on that machine, or lower the internet access rules.

Limit of 100MB of image files within a single job

The browser component (Mozilla Gecko) used in the WYSIWYG editor of the Designer was updated for Connect 2018.1. This allows use of new CSS properties, such as flexbox.

However this update also introduced a limit of 100MBs for image files included within a single job. The limit is set at 100MB deliberately, as this allows *most* jobs to run faster. However, if a job requires more than 100MBs of image files, then the Connect image cache size can be increased to cater for such.

Please contact **OL Support** for instructions on how to modify the image memory cache value, if needed.

Print Output: Booklet Impositioning changes introduced in 2018.1

When Booklet Impositioning is enabled, all pages within a document need to be changed to duplex prior to Impositioning . The method for duplexing jobs has been changed to now always combine existing pages into the front and backsides of sheets, rather than adding empty backsides to any simplex pages.

The result is that now every document in the job becomes a booklet without any empty pages between the first page and the last page.

With some exceptions. Booklet Impositionings that require a multiple of 4 pages (*Saddle binding* and *Perfect binding*) will still get empty pages added, when needed.

Installation paths with multi-byte characters

When installing the Traditional Chinese or Japanese versions of Connect, if the user specifies an alternative installation path containing multi-byte/wide-char characters it can break some of the links to the Connect-related shortcuts in the Start Menu and cause an error to appear at the end of the installer. The workaround for the moment is to use the default installation path. The problem will be addressed in a later release.

Switching languages

Changing the language using the **Window > Preferences > Language Setting** menu option does not currently change all of the strings in the application to the selected language. This is a known issue and will be fixed in a later release.

In the meantime we offer the following workaround for anyone who needs to change the language:

1. Go to the .ini files for the Designer and Server Config:
    - `C:\Program Files\Objectif Lune\OL Connect\Connect Designer\Designer.ini`
    - `C:\Program Files\Objectif Lune\OL Connect\Connect Server Configuration\ServerConfig.ini`
2. Change the language parameter to the required one under Duser.language=en | es | de | fr | it | ja | ko | pt | tw | zh

Only one of the above language tags should be selected. Once saved, Connect will appear in the selected language at next start-up.

GoDaddy certificates

When installing Connect offline, dialogs allow installing the GoDaddy certificates. Most users should use the default settings and click **Next**. In some cases, however, this may not work correctly. For this reason those users should activate **Place all certificates in the following store** and then select the **Trusted Root Certification Authorities** as the target certificate store.

MySQL Compatibility

The minimum supported MySQL version is MySQL 5.6.

Available Printer Models

Note that only the single Printer Model (Generic PDF) will appear on the **Advanced** page of the **Print Wizard** by default.

To add additional printer models click on the settings button next to the Model selection entry box.

Color Model in Style Sheets

The color model of colors defined in a style sheet can sometimes change after editing the style sheet. This is a known issue and will be addressed in a subsequent release.

Image preview in Designer

If in the Windows Internet settings (**Connection Settings > LAN configuration**) a proxy is enabled, but "Bypass proxy settings for local addresses" is not checked, the image preview service, conversion service and live preview tab in the Designer will not work and exhibit the following issues:

- Images will be shown as 0 size boxes (no red 'X' is displayed).
- Live preview does not progress, and when re-activated reports "browsers is busy".

To fix the issue you must check the "Bypass proxy settings for local addresses" option.

VIPP Output

Some templates set up with landscape orientation are being produced as portrait in VIPP. It can also sometimes be the case that text and images can be slightly displaced. These are known issues and will be addressed in a later release of Connect.

Magic Number changes when installing Docker

Installing Docker on a system where Connect has already been installed may impact Connect's licensing mechanism and require reactivation.

Note

Installing Connect after Docker has already been installed will not cause issues.

Uninstalling

This topic provides some important information about uninstalling (removing) PrintShop Mail Connect2022.1.

To uninstall PrintShop Mail Connect select the application from within the Add/Remove programs option under the Control Panel. This will start the **PrintShop Mail Connect Setup Wizard** in uninstall mode.

Note

The **PrintShop Mail Connect Setup Wizard** might take some seconds to appear.

Important: Stop any active Anti-Virus software before uninstalling Connect back-end database.

Some anti-virus systems are known to block the uninstallation of MariaDB datafiles, as well as blocking the uninstallation of the MariaDB database application itself. If you wish to uninstall the Connect back-end database it is **highly recommended** that any anti-virus application be stopped prior to uninstalling PrintShop Mail Connect , as otherwise the Connect uninstallation might not work correctly.

Impacts upon other applications and services

- The Uninstall will terminate the installed Server / MariaDB service(s).
- The following applications / services should be stopped in a controlled fashion, before running the PrintShop Mail Connect Uninstall:
    1. PrintShop Mail Connect
    2. Connect products on remote systems which refer to this MariaDB database.

Uninstallation Wizard

The uninstallation is done by running the PrintShop Mail Connect Setup Wizard in uninstall mode. The Wizard contains the following important pages:

1. **PrintShop Mail Connect Setup**: This page allows selection of what is to be done. An modification to the existing installation (**Add or Remove Features**) or full Uninstall.
   NOTE: If the Uninstall option is selected the **Remove user data** option is made available. For information about exactly what data would be saved or deleted, please see "Upgrading from Connect 2021.2 or earlier with Remove User Data CHECKED:" on page 57.
2. **Component Selection**: If **Add or Remove Features** was selected on previous screen, this page provides options for adding or removing Connect features.
   For detail descriptions of the options, see Installation "Component Selection" on page 33
   .

Note

If an error occurs during uninstallation or after/when re-installing Connect after uninstalling it, please see: Problems during a Connect installation or version upgrade in Connect's Knowledge Base (https://help.objectiflune.com/en/kb-connect/#KB/FAQ/General/Installation_Upgrade.htm).

General information

Connect consists of visible and invisible parts. The invisible parts process the Connect job to provide the actual output. They are introduced to you in the topic: "Connect: a peek under the hood" below.

For information about Connect logging, see "Log files" on page 104.

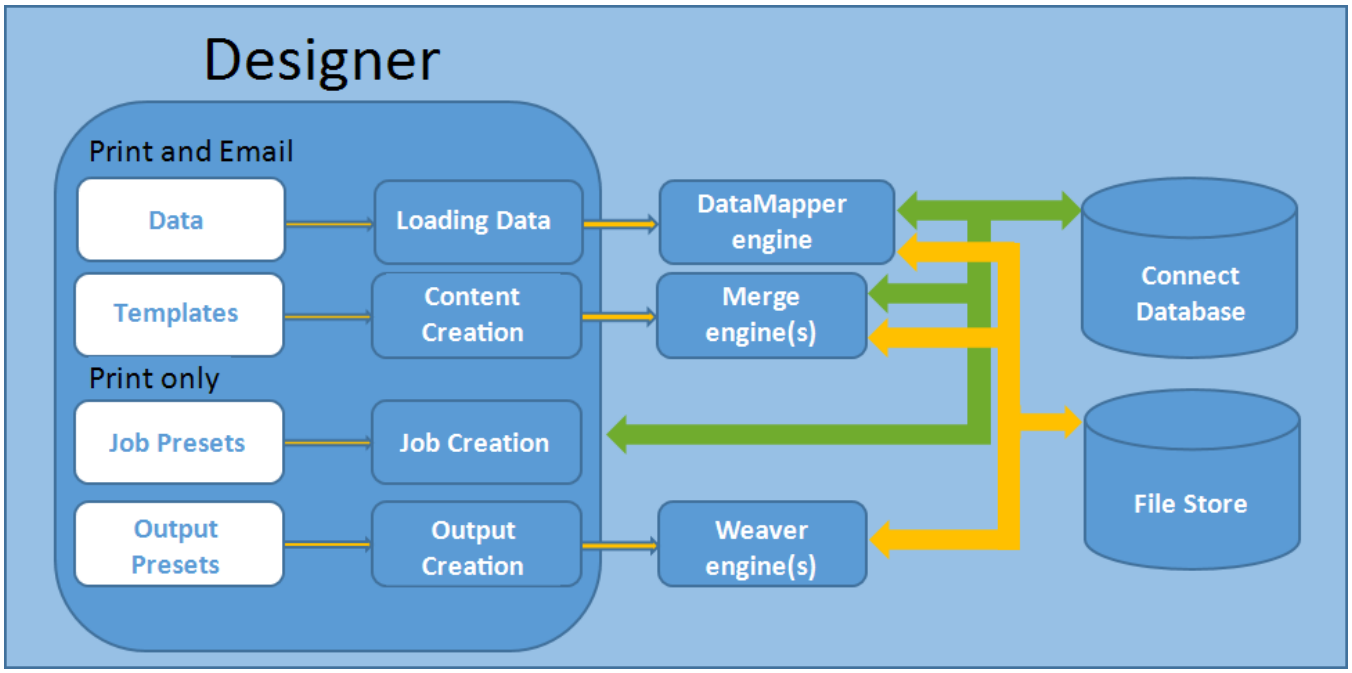
For a list of all file types used in Connect, see: "Connect file types" on page 105.

You can find additional information that complements the user manuals, such as error codes and frequently asked questions about PrintShop Mail Connect, in the Knowledge base.

Connect: a peek under the hood

Connect consists of visible and invisible parts.
The visible part is of course the Designer, that you use to create templates and Print Presets.
The invisible parts process the Connect job to provide the actual output. This topic introduces you to those parts.

Here's a simplified, graphical representation of the architecture of PrintShop Mail Connect.



The Connect database

The Connect database is the database back-end used by Connect itself when processing jobs. It can be either the MariaDB instance provided by the Connect installer, or a pre-existing (external) instance (see "Database Considerations" on page 15).

All generated items (records, content items etc.) are stored in this database. .

> **Note**
>
> Email content items are not stored in the Connect database.

A clean-up of the database is performed at regular intervals in accordance with the settings (see "Clean-up Service preferences" on page 370).

The File Store

Connect has its own File Store which it uses for transient files.
The Clean-up service takes care of removing obsolete files when those files are not marked as permanent (see "Clean-up Service preferences" on page 370).

The engines

DataMapper engine. A DataMapper engine extracts data from a data file. This is an internal engine, for which there are no settings to make.

Merge engine/s. A merge engine merges data with a template using the scripts in the template, in order to create content items.
The number of merge engines is configurable (see "Engines preferences" on page 84): it can be increased depending on the capacity of the machine that runs the solution (see "Performance Considerations" on page 23).

Weaver engines. A Weaver engine creates Print output from Print content items. It takes the settings made in Print Presets or in the Print Wizard into account.
The number of Weaver engines is configurable as well (see "Engines preferences" on page 84).

Log files

Location

Connect's log files are located in the following folder: `C:\Users\[UserName]\Connect\logs`.
Where `[username]` is replaced by the appropriate Windows user name.

> **Tip**
>
> Actually, the path may not begin with 'C:\Users', as this is language-dependent. On a French system, for example, it would be 'C:\Utilisateurs'.
>
> Type `%userprofile%` in a Windows File Explorer and press Enter to open the actual current user's home directory.

Every time output is generated, the Designer and any engines involved in the operation produce their own log files.
Each component writes its log files to a dedicated subfolder of the Log folder. Merge engines write to the `logs/Mergeengine` folder, Weaver engines to the `logs/Weaverengine` folder and so forth.

Note that actions of the Cleanup service are only logged in the Server's log file. (See also: "Clean-up Service preferences" on page 370.)

> **Tip**
>
> For more information about Connect's architecture, see: "Connect: a peek under the hood" on page 102.

Name

The name of a log file consists of the **component**'s name, a **time stamp** and the Windows **process ID**.
The Windows process ID is needed since multiple engines of the same type could be running at the same time, depending on the engine configuration (see Engine configuration) and the size of a job.

Format

The log files are **text** files that can be opened in any text processor. Each log file contains a header section containing a snapshot of the current working environment (operating system, Connect version, system locale settings, disk and RAM information, and such like).

The logging **level** is configurable, as well as the exact **format** - the "logging pattern" - of the individual log messages.

- The Designer's logging preferences are set via the "Logging preferences" on page 387.

The Designer's log messages are also displayed in the Messages pane (see "Preflight Results and Messages" on page 581). The Log Filter allows to define what kind of events may be shown in the Messages pane (see Log Filter). That doesn't change the actual logging level of the Designer, however; it only filters the log file.

Connect file types

This article describes the different file types that are related to PrintShop Mail Connect and its different modules. These are files that are generally transferable between machines, can be sent via email or other means.

- **.OL-template**: A Designer Template file (see "Templates" on page 109). Is linked to a data mapping configuration by default, but not necessarily.
- **Print Presets** are used when generating a job, ready for output, from Designer (see "Generating Print output" on page 993) or through Workflow (see Print processes with OL Connect tasks).
    - **.OL-jobpreset**: A Job Creation Preset file has settings for data filtering and sorting, grouping (groups are used to split the print output), adding metadata fields, and finishing options that override the template's Media and Section settings. (See "Job Creation Presets" on page 999.)
    - **.OL-outputpreset**: An Output Creation Preset file is used to generate the actual print output in the appropriate format (pcl, pdf, etc). It includes print settings such as impositioning (N-up, cut & stack), inserter marks, tray settings, separation etc. (See "Output Creation Presets" on page 1001.)
- **.OL-package**: A transfer file used to package one or many of the above files (the data model being part of both the template and the data mapping configuration). Created by using the **File > Package** dialog. (See "Package dialog" on page 520.)
- **.OL-script**: One or more Designer scripts. Scripts personalize the output of a template. They are either added via wizards (see "Personalizing content" on page 338) or self-written (see "Writing your own scripts" on page 393). They can also be imported or exported from the Scripts pane in Designer when a template is open.
- **.OL-printerdef**: A Printer Definition File is used by the Output Creation Preset (see "Output Creation Presets Wizard" on page 700) to determine what type of output to produce. These definition files are generated by an internal application that is not currently distributed outside of OL, but the definition files themselves can be provided.

The Designer

The Designer is a WYSIWYG (what you see is what you get) editor that lets you create templates for Print and Email output.

A template may contain designs for multiple output channels: a letter intended for print and an e-mail variant of the same message, for example. Content, like the body of the message or letter, can be shared across these contexts.

Templates are personalized using scripts and variable data. More advanced users may edit the underlying HTML, CSS and JavaScript directly.

The following topics will help to quickly familiarize yourself with the Designer.

- "Designer basics" below. These are the basic steps for creating and developing a template.
- "Features" on the facing page. These are some of the key features in the Designer.
- "Designer User Interface" on page 476. This part gives an overview of all elements in the Designer User Interface, like menus, dialogs and panes.

More help can be found here:

- Tutorials On Video: watch an introductory video, overview tutorials or practical how-to videos.
- Forum: Browse the forum and feel free to ask questions about the use of Connect software
- Demo site. Download demonstrations of OL products.

Designer basics

With the Designer you can create templates for personalized letters and emails, and generate output from them.

These are the basic steps for creating and developing a template:

1. **Create a template**
   Create a template, using one of the Template Wizards. See "Creating a template" on

page 109.

2. **Fill the template**
   Add text, images and other elements to the template and style them. See "Content elements" on page 191 and "Styling and formatting" on page 294.
3. **Personalize the content**
   Personalize the content using variable data. See "Personalizing content" on page 338.
4. **Generate output**
   Adjust the settings, test the template and generate output: letters andemails. See "Generating output" on page 992.

Note that steps 2 and 3 are not necessarily to be followed in this order. For example, as you add elements to a template, you may start personalizing them right away, before adding other elements to the template.

Features

The Designer is Connect's module to create templates for personalized customer communications. These are some of the key features in the Designer:

"Templates" on the next page. Start creating, using and sharing templates.

"Contexts" on page 118. A context contains one or more designs for one output channel:

- "Print" on page 124. This topic helps you design and fill sections in the Print context.
- "Email" on page 166. This topics helps you design an email template.
- Web. This topic helps you design a web page.

"Sections" on page 119. Sections in one context are designed for the same output channel.

"Content elements" on page 191. Elements make up the biggest part of the content of each design.

"Snippets" on page 285. Snippets help share content between contexts, or insert content conditionally.

"Styling and formatting" on page 294. Make your Designer templates look pretty and give them the same look and feel with style sheets.

"Personalizing content" on page 338. Personalize your customer communications using variable data.

"Writing your own scripts" on page 393. Scripting can take personalization much further. Learn how to script via this topic.

"Generating output" on page 992. Learn the ins and outs of generating output from each of the contexts.

Templates

The Designer is a WYSIWYG (what you see is what you get) tool to create templates. This topic gets you started. It explains how to create a template, what is found in a template file, and how output can be generated.

Creating a template

In the **Welcome** screen that appears after startup, get off to a flying start choosing **Template Wizards**. Scroll down to see all the Template Wizards. After deciding which output channel will be prevalent in your template, select a template.

The Template Wizards can also be accessed from the menu: click **File**, click **New**, expand the **Template** folder, and then expand one of the **templates** folders.

There are Wizards for each output channel, or **contexts** as they are called in the Designer: Print and Email.
See:

- "Creating an Email template with a Wizard" on page 171
- "Creating a Print template with a Wizard" on page 126

> **Tip**
>
> The quickest way to create a Print template based on a PDF file is to **right-click the PDF** file in the Windows Explorer and select **Enhance with Connect**.

After creating a template you can add the other contexts (see "Contexts" on page 118), as well as extra sections (see "Sections" on page 119), to the template.

It is, however, not possible to use a Template Wizard when adding a context or section to an existing template.

> **Tip**
>
> When you add an Email context to an existing template you get a 'basic action email'. This is one of the 4 types of email that you can choose from when you start a template with an Email Template Wizard; see "Creating an Email template with a Wizard" on page 171.

Opening a template

To open a template from the Welcome screen, select **Open File**.
To open a template from the menu, select **File > Open**.
Then select the template file. A template file has the extension .OL-template.

The most recently opened templates and data mapping configurations are listed on the Welcome screen and in the menu: **File > Open Recent**.
To clear these lists, select **Window > Clear Recent Files Lists**.

> **Warning**
>
> A template created in an older version of the software can be opened in a newer version. However, opening and saving it in a newer version of the software will convert the template to the newest file format. The converted template can't be opened in older versions of the software.

Opening a package file

Templates can also be stored in a package file (see "Creating package files" on page 113). To open a package file, switch the file type to Package files (*.OL-package) in the Open File dialog. If the package contains Print Presets, you will be asked if you want to import them into the proper repositories.

Saving a template

A Designer template file has the extension **.OL-template**. It is a zip file that includes up to 3 contexts, all the related resources and scripts, and (optionally) a link to a data mapping

configuration.

To save a template, select **File > Save** or press **Ctrl+S**. The first time you'll have to give the template a name. **File > Save as** allows you to save the template with a different name. The **Save as** dialog also appears when saving a template that is read-only.

> **Tip**
>
> To quickly copy the name of any other file, set **Save as type** to **Any file (*.*)** in the Save dialog. Select a file to put its name in the File name field. Then set **Save as type** to **Template files (*.OL-template)** and save the template.

When more than one resource is open and the Designer software is closed, the Save Resources dialog appears. This dialog displays a list of all open resources with their names and file location. Selected resources will be saved, deselected resources will have all their changes since they were last saved dismissed.

Saving older templates

Saving a template in a newer version of the software will convert the template to the newest file format. This makes it unreadable to older versions of the software.
The warning message that is displayed in this case can be disabled.
To re-enable this message (and all other warning dialogs), go to **Window > Preferences > General**, and click the **Reset All Warning Dialogs** button at the bottom.

Read-only templates

A lock icon displayed next to the file name indicates that the file is read-only. Saving a read-only file opens the Save as dialog, allowing you to save the file under a different name. The new file is then opened and can be edited in the Designer.

Auto Save

After a template has been saved for the first time, Connect Designer can auto save the template with a regular interval. To configure Auto Save:

1. Select the menu option **Window > Preferences > Save**.
2. Under **Auto save**, check the option **Enable** to activate the Auto Save function.
3. Change how often it saves the template by typing a number of minutes.

Auto Backup

Connect Designer can automatically create a backup file when you **manually** save a template. To configure Auto Backup:

1. Select the menu option **Window > Preferences > Save**.
2. Under **Auto backup**, check the option **Enable** to activate the Auto Backup function.
3. Type the number of revisions to keep.
4. Select the directory in which the backups should be stored.

Backup files have the same name as the original template with two underscores and a progressive number (without leading zeros) at the end: **originalname__1.OL-template**, **originalname__2.OL-template**, etc.

> **Note**
>
> The Auto Save function does **not** cause backup files to be created.

File properties

On the menu, select **File > Properties** to view and complement the file properties. See "File Properties dialog" on page 496.
The file properties can also be used in scripts; see "template" on page 960. If you are not familiar with writing scripts, refer to "Writing your own scripts" on page 393.

Saving a copy / down-saving a template

The Connect software is backwards compatible: templates that were made with an older version of Connect can always be opened with the newest version of the software.
But newer templates cannot be opened with an older version of the software.
Connect, however, allows you to save a template in the format of a previous version of the software, so that you may restore a template that was accidentally opened and saved in a newer version, or share a template with users of a previous version of Connect.
Note that it may not always be possible to down-save a template. If a template uses a feature that did not exist in a certain older version, it won't be possible to down-save it to that older version.

To **down-save** a template, select **File > Save a Copy**, from the menu. Select the software version and click OK. Next you can select a location and give the template a name, as usual.

To **save a copy**, follow the same procedure without selecting a previous version of the software.

Creating package files

A package file (*.OL-package) contains one or more templates and Print Presets.

Print Presets make it possible to do such things as filtering and sorting records, grouping documents and splitting the print jobs into smaller print jobs, as well as the more standard selection of printing options, such as binding, OMR markings and the like. See "Print Presets" on page 998 for more details.

Package files can be opened by other Connect users.

To open the Package dialog, select **File > Package...**. For an explanation of the options in the Package dialog, see "Package dialog" on page 520.

Generating output from the Designer

Output can be generated directly from the Designer; see "Generating Print output" on page 993, and "Generating Email output" on page 1017.

To test a template first, select **Context > Preflight**. Preflights execute the template without actually producing output and it displays any issues once it's done (see also: "Testing scripts" on page 403).

Exporting a template report

A template report can be used for archiving purposes or to provide information about the template to people who do not have access to Connect. Such a report can be exported in PDF or XML format. By default it contains a summary of the template with an overview of all the settings and resources that are used in the template: media, master pages, contexts, sections, images, scripts, runtime parameters, etc. The file properties are included as well (see "File Properties dialog" on page 496).

To open the Export Template Report wizard, select **File > Export Report**. For a description of all options, see "Export Template Report wizard" on page 495.

Creating a custom template report

The Export Template Report wizard also offers the possibility to export custom template reports (in PDF format only). A custom template report could contain another selection of information and present that differently, e.g. with the logo of your company.

To create a custom template report, you need two files: A template design with the desired layout and variable data. This .OL-template file has to be made in the Designer. A data mapping configuration provides the variable data. You could use the data mapping configuration made for the standard template report, or create another one in the DataMapper module, using the standard XML template report as data sample.The DataMapper is included only in PlanetPress Connect and PReS Connect. Data mapping configurations have the extension .OL-datamapper. The following zip file contains both the template and data mapping configuration that are used to generate the standard template report: http://help.objectiflune.com/en/archive/report-template.zip.

Creating a Word-Based Print template with mail merge

If your Word document does not include mail merge fields, see "Creating a Print template with a Wizard" on page 126.

There are three general steps required when you want to import a Word file that includes mail merge fields:

- "Create a Word document that includes mail merge fields" below
- "Import the Word Mail Merge document as a template " on the next page
- "Add the data source to create the data model" on the next page

Create a Word document that includes mail merge fields

If you do not already have a mail merge document and data file, you will need to create one in Word. Also, you need to have your data file in a format supported by Word.

1. Create or open a Word document.
2. Enter the content for the document, including placeholders for the data you want to use.
3. If you already know what your data fields are called you can enter them directly. You can also derive them from the data source by selecting **Mailings**, then **Select Recipients > Use an Existing List** .

4. Select the data file that you want to use. Word will attempt to automatically determine the file type and select appropriate options, but you can adjust the parameters if needed.
5. As you create your Word document, you insert the fields using **Insert Merge Fields**.
6. Save the file.

Import the Word Mail Merge document as a template

1. From within Connect Designer, select **File > New** and then select the **Word-based Print** template.
2. Navigate to the Word document, select it, then select **Finish** to import the file. Note that the mail merge fields are automatically converted to the proper format for Designer. Images from the word file are in the Images folder, and a script is created for each of the fields detected in the file.
3. Modify the template as needed. You may want to place imported images on the master page, add headers or footers, etc.
4. Continue with importing the data source.

Add the data source to create the data model

1. Select **File > Add data > From File Data Source** and navigate to the data source.
2. Select the file. The wizard displays a preview of the data with the attributes of the file that were automatically detected. Adjust the attributes if necessary.
3. Select **Finish** to import the data. At this point you can use the Preview tab to review the template with the data in place.

Resources

This page clarifies the difference between Internal, External and Web resources that may be used in a template, and explains how to refer to them in HTML and in scripts.

All resources that a template uses are displayed in the "Resources pane" on page 583.

Tip

The Text Filter box at the top of the Resources pane allows to look for text in the name of resources and in the source of any text-based files: HTML (including sections, master pages, and snippets), JSON, JS and CSS.

Internal resources

Internal resources are files that are added to and saved with the template.

To add images, fonts, style sheets, and snippets to your template, you can **drag** or **copy/paste** them into the Resources Pane. See also: "Images" on page 271, "Snippets" on page 285, "Styling templates with CSS files" on page 296 and "Fonts" on page 332.
Resource files can also be dragged or copy/pasted **out** of the the application to save them on a local hard drive.

Alternatively you could **import** resources and scripts from another template; click **File > Import Resources...** in the menu and select a template to import resources from (see).

Once added or imported, internal resources are accessed using a relative path, depending where they're called from. Resources can be located in the following folders:

- images/ contains the files in the Images folder.
- fonts/ contains the files in the Fonts folder.
- css/ contains the files in the Stylesheets folder.
- js/ contains the files in the JavaScripts folder.
- snippets/ contains the files in the Snippets folder.

When refering to them, normally you would simply use the path directly with the file name. The structure within those folders is maintained, so if you create a "signatures" folder within the "Images" folder, you need to use that structure, for example in HTML: <img src="images/signatures/johnsmith.gif">. In scripts, you can refer to them in the same way, for example:

```
results.loadhtml("snippets/en/navbar.html");
```

See also: "Loading a snippet via a script" on page 420 and "Writing your own scripts" on page 393.

Note

When referring to images or fonts from a CSS file, you need to remember that the current path is css/, meaning you can't just call images/image.jpg. Use a relative path, for example: #header { background-image: url('../images/image.jpg'); }

External resources

External resources are not stored in the template, but on the local hard drive or on a network drive. They are accessed using a path. The path must have forward slashes, for example <img src="file:///c:/resources/images/signatures/johnsmith.gif"> or var json_variables = loadjson ("file:///d:/jsondata/variables.json");. The complete syntax is: file://<host>/<path>. If the host is "localhost", it can be omitted, as it is in the example, resulting in **file:///<path>**. The empty string is interpreted as `the machine from which the URL is being interpreted'.

Network paths are similar: results.loadhtml("file://servername/sharename/folder/snippet.html"); (note that in this case **file** is followed by 2 slashes only).

Some limitations

- Style sheets cannot refer to external resources.
- The Connect Server user needs access to whichever network path is used. If the network path is on a domain, the Connect Server must be identified with domain credentials that have access to the domain resources.

For more information on network paths, please see this Wikipedia entry: file URI scheme.

Web resources

Web resources are simply accessed using a full URL. This URL needs to be publicly accessible: if you type in that URL in a browser on the server, it needs to be visible. Authentication is possible only through URL Parameters (http://www.example.com/data.json?user=username&password=password) or through HTTP Basic Auth (http://username:password@www.example.com/data.json).

Contexts

Contexts are parts of a template that are each used to generate a specific type of output: Email or Print.

- The Print context outputs documents to either a physical printer or a PDF file; see "Print context" on page 132.
- The Email context outputs HTML email, composed of HTML code with embedded CSS. See "Email context" on page 174.

When a new template is made, the Context appropriate to that new template is automatically created, including one section. After a template has been created, the other two contexts can be added to it; see "Adding a context" on the next page.

> **Tip**
>
> When you add an Email context to an existing template you get a 'basic action email'. This is one of the 4 types of email that you can choose from when you start a template with an Email Template Wizard; see "Creating an Email template with a Wizard" on page 171.

Outputting and combining contexts

All contexts can be present in any template and they can all be used to output documents; see "Generating Email output" on page 1017, "Generating Print output" on page 993 and Generating Web output.

They can even be combined in output.

If present in the same template, a Print context can be attached to an Email context.

You could select Print sections based on a value in the data with a Conditional Print Sections script; see "Conditional Print sections" on page 363.

Outputting other combinations of contexts, and selecting other sections based on a value in the data, can be done via a Control Script; see "Control Scripts" on page 429.

Adding a context

To add a context, right-click the **Contexts** folder on the **Resources** pane and click **New print context**, **New email context** or **New web context**. Or use **Context > Add** in the main menu. Only one context of each type can be present in a template. Each context, however, can hold more than one section; see "Sections" below.

Importing a context

To import a context, click **File > Import Resources...** in the menu. See: "Import Resources dialog" on page 502.

Deleting a context

To delete a context, right-click the context on the **Resources** pane and click **Delete**.

> **Warning**
>
> If you don't have a backup of the template, the only way to recover a deleted section, is to click **Undo** on the **Edit** menu, until the deleted section is restored. After closing and reopening the template it is no longer possible to restore the deleted context this way.
> In the Saving Preferences you can set whether a backup file should be created when you save the template; see "Save preferences" on page 391.

Sections

Sections are parts of one of the contexts in a template: Print or Email.

They contain the main text flow for the contents. In each of the contexts there can be multiple sections. A Print context, for example, may consist of two sections: a covering letter and a policy.

Adding a section

To add a section to a context, right-click the context (Email or Print) on the **Resources** pane, and then click **New section**.

The new section has the same settings as the currently active section in the same context, or the first section in the same context if another context is active.

It is not possible to use a Template Wizard when adding a section to an existing template.

> **Tip**
>
> When you add an Email context to an existing template you get a 'basic action email'. This is one of the 4 types of email that you can choose from when you start a template with an Email Template Wizard; see "Creating an Email template with a Wizard" on page 171.

Importing a section

To import a section from another template, click **File > Import Resources...** in the menu. See: "Import Resources dialog" on page 502.
Remember to copy the related source files, such as images, to the other template as well.

Editing a section

To open a section, expand the **Contexts** folder on the **Resources** pane, expand the respective context (**Print** or **Email**) and double-click a section to open it.

Each section can contain text, images and many other elements (see "Content elements" on page 191), including variable data and other dynamic elements (see "Personalizing content" on page 338).

To preview a section, open the Preview tab in the Workspace (see "Workspace" on page 595).

Copying a section

To copy a section:

1. Open the context on the **Resources** pane.
2. Right-click the section and select **Copy**.
3. Right-click the context and select **Paste**. Note that sections cannot be pasted to another context.

The copy will have the same settings as the original.

Alternatively you may copy the contents of a section manually:

1. Open the section that you want to copy and go to the **Source** tab in the workspace.
2. Copy the contents of the **Source** tab (press **Ctrl+A** to select everything and then **Ctrl+C** to copy the selection).
3. Add a new section (see "Adding a section" on page 119, above).
4. Go to the **Source** tab and paste the contents of the other section here (press **Ctrl+V**).
5. When copying a section to another template, add the related source files, such as images, to the other template as well.

> **Tip**
>
> The easiest way to copy a section to another template, is to use the **Import Resources** dialog in the other template. See: "Import Resources dialog" on page 502.

Deleting a section

To delete a section:

- On the **Resources** pane, expand the **Contexts** folder, expand the folder of the respective context, right-click the name of the section, and then click **Delete**.

> **Warning**
>
> No backup files are maintained in the template. The only way to recover a deleted section, is to click **Undo** on the **Edit** menu, until the deleted section is restored, or by reverting to the last saved state (click File > Revert, on the menu). After closing and reopening the template it is no longer possible to restore the deleted context this way.

Renaming a section

To rename a section:

- On the **Resources** pane, expand the **Contexts** folder, expand the folder of the respective context, right-click the name of the section, and then click **Rename**.

Section properties

Which properties apply to a section, depends on the context it is part of. See also: "Print sections" on page 137, "Email templates" on page 177, and Web pages.

To change the properties for a section:

1. On the **Resources** pane, expand the **Contexts** folder.
2. Expand the folder of the respective context.
3. Right-click the name of the section, and then click one of the options.

Applying a style sheet to a section

In order for a style sheet to be applied to a specific section, it needs to be included in that section. There are two ways to do this.

Drag & drop a style sheet

1. Click and hold the mouse button on the style sheet on the **Resources** pane.
2. Move the mouse cursor within the **Resources** pane to the section to which the style sheet should be applied.
3. Release the mouse button.

Using the Includes dialog

1. On the Resources pane, right-click the section, then click **Includes**.
2. From the **File types** dropdown, select **Stylesheets**.
3. Choose which CSS files should be applied to this section. The available files are listed at the left. Use the arrow buttons to move the files that should be included to the list at the right.
4. You can also change the order in which the CSS files are read: click one of the included CSS files and use the **Up** and **Down** buttons. Note that moving a style sheet up in the list gives it **less** weight. In case of conflicting rules, style sheets read later will override previous ones.

Note

Style sheets are applied in the order in which they are included in a section. The styles in each following style sheet add up to the styles found in previously read style sheets. When style sheets have a conflicting rule for the same element, class or ID, the **last** style sheet 'wins' and overrides the rule found in the previous style sheet.

Note

Style sheets that are linked to (i.e. included in) a section show a chain icon in the Resources pane (see "Resources pane" on page 583).

Arranging sections

Changing the order of the sections in a context can have an effect on how they are outputted; see: "Print sections" on page 137, "Email templates" on page 177 and Web pages.

To rearrange sections in a context:

- On the **Resources** pane, expand the Contexts folder, expand the folder of the respective context, and then drag and drop sections to change the order they are in.
  Alternatively, right-click a section and click **Arrange**. In the Arrange Sections dialog you can change the order of the sections in the same context by clicking the name of a section and moving it using the **Up** and **Down** buttons.

Outputting sections

Which sections are added to the output, depends on the type of context they are in.

When generating output from the Print context, each of the Print sections is added to the output document, one after the other in sequence, for each record. The sections are added to the output in the order in which they appear on the **Resources** pane. See "Generating Print output" on page 993.

In Email output, only one section can be outputted at a time. The section that will be output is the section that has been set as the 'default'. See "Generating Email output" on page 1017 and "Email templates" on page 177.

It is, however, possible to include or exclude Print sections when the output is generated, depending on a value in the data. A Control Script can do this; see "Control Scripts" on page 429.

See "Generating output" on page 992 to learn how to generate Print documents or Email.

Print

Connect supports a number of different types of print outputs. These include:

- AFP
- IPDS (spool file created by Connect, with printer control available through the "Print Manager" on page 1029)
- PCL
- PDF
- PostScript (including the PPML, VIPP and VPS variants)

With the Designer you can create one or more Print templates and merge the template with a data set to generate personal letters, invoices, policies, or any other type of letter you can think of.

The Print **context** is the folder in the Designer that can contain one or more Print sections.

Print templates (also called Print *sections*), are part of the Print context. They are meant to be printed directly to a printer or a printer stream/spool file, or to a PDF file (see "Generating Print output" on page 993).
The Print context can also be added to Email output as a PDF attachment; see "Generating Email output" on page 1017.
When generating output from the Print context, each of the Print sections is added to the output document, one after the other in sequence, for each record.

When a Print template is created or when a Print context is added to an existing template the Print context folder is created along with other folders and files that are specific to a Print context (see "Creating a Print template with a Wizard" on page 126, "Adding a context" on page 119 and "Print context" on page 132).

Only one Print section is created at the start, but you can add as many Print sections as you need; see "Print sections" on page 137.

Pages

Unlike emails, Print sections can contain multiple *pages*. Pages are naturally limited by their size and margins. If the content of a section doesn't fit on one page, the overflow goes to the next page. This happens automatically, based on the section's page size and margins; see "Page settings: size, margins and bleed" on page 148.
The minimum number of pages can be set via the Print section properties; see "Print section properties" on page 542.

Although generally the same content elements can be used in all three contexts (see "Content elements" on page 191), the specific characteristics of pages make it possible to use special elements, such as page numbers; see "Page numbers " on page 150.

See "Pages" on page 147 for an overview of settings and elements that are specific for pages.

Headers, footers, tear-offs and repeated elements (Master page)

In Print sections, there are often elements that need to be repeated across pages, like headers, footers and logos. In addition, some elements should appear on each first page, or only on pages in between the first and the last page, or only on the last page. Examples are a different header on the first page, and a tear-off slip that should show up on the last page.

This is what Master Pages are used for. Master Pages can only be used in the Print context.

See "Master Pages" on page 156 for an explanation of how to fill them and how to apply them to different pages.

Stationery (Media)

When the output of a Print context is meant to be printed on paper that already has graphical and text elements on it (called stationery, or preprinted sheets), you can add a copy of this media, in the form of a PDF file, to the Media folder.

Media can be applied to pages in a Print section, to make them appear as a background to those pages. This ensures that elements added to the Print context will correspond to their correct location on the preprinted media.

When both Media and a Master Page are used on a certain page, they will both be displayed on the Preview tab of the workspace, the Master Page being 'in front' of the Media and the Print section on top. To open the Preview tab, click it at the bottom of the Workspace or select **View > Preview View** on the menu.

The Media will not be printed, unless this is specifically requested through the printer settings in the Print Wizard; see "Generating Print output" on page 993.

See "Media" on page 159 for further explanation about how to add Media and how to apply them to different pages.

Copy Fit

Copy Fit is a feature to automatically adjust the font size of text to make it fit the available space. It could be used for the name of a person on a greeting card, for instance, or for the name of a product on a shelf talker. This feature is only available with Box and Div elements in Print sections.

For more information about this feature see "Copy Fit" on page 311.

Creating a Print template with a Wizard

A Print template may consist of various parts, such as a covering letter and a policy. Start with one of the Template Wizards for the first part; other parts (called 'sections') can be added later.

Print template wizards can be found in the Welcome screen and on the File menu.

In the **Welcome** screen that appears after startup:

- Choose **Template Wizards** and scroll down until you see the **Basic Print templates** or **ERP templates** and select one of them.
- Or choose **New Template** and select **Print**, **PDF-based Print**, or **Word-based Print**.

Alternatively, on the menu select **File > New**, expand the **Template** folder, and then:

- Select **PDF-based Print** or **Word-based Print**.
- Or expand the **Basic Print templates** or **ERP templates** folder, select a template type and click **Next**.

Another way to start a PDF-based Print template is this:

- **Right-click the PDF** file in the Windows Explorer and select **Enhance with Connect**.

The various template types and their options are described below.

See "Print context" on page 132 and "Print sections" on page 137 for more information about Print templates.

> **Tip**
>
> Use the **Outline** pane at the left to see which elements are present in the template and to select an element.
>
> Use the **Attributes** pane at the right to see the current element's ID, class and some other properties.
>
> Use the **Styles** pane next to the Attributes pane to see which styles are applied to the currently selected element.

Basic Print template wizards

There are two 'basic' Print Template wizards: one for a formal letter, and one for a postcard.

Postcard

The Postcard Wizard lets you choose a page size and two background images, one for the front and one for the back of the postcard.

When you click **Finish**, the Wizard creates:

- A Print context with one section in it, that has duplex printing (printing on both sides) enabled. See "Printing on both sides" on page 135.
- Two Master Pages that each contain a background image. The first Master Page is applied to the front of every page in the Print section. The second Master Page is applied to the back of every page in the Print section. See "Master Pages" on page 156.
- **Scripts** and **selectors** for variable data. The **Scripts** pane shows, for example, a script called "first_name". This script replaces the text "@first_name@" on the front of the

postcard by the value of a field called "first_name" when you open a data set that has a field with that name. See "Variable Data" on page 351.

- A script called Dynamic Front Image Sample. This script shows how to toggle the image on the front page dynamically. See also "Writing your own scripts" on page 393.
- One empty Media. Media, also called Virtual Stationery, can be applied to all pages in the Print section. See "Media" on page 159.

The Wizard opens the Print section, so that you can fill it with text and other elements; see "Content elements" on page 191. It already has two Positoned Boxes on it: one on the front, for text, and one on the back, for the address.

See "Print context" on page 132 and "Print sections" on page 137 for more information about Print templates.

Formal letter

The Formal Letter Wizard first lets you select the page settings, see "Page settings: size, margins and bleed" on page 148.

These settings are fairly self-explanatory, except perhaps these:

- Duplex means double-sided printing.
- The margins define where your text flow will go. The actual printable space on a page depends on your printer.
- The bleed is the printable space **around** a page. It can be used on some printers to ensure that no unprinted edges occur in the final trimmed document. Printers that can’t print a bleed, will misinterpret this setting. Set the bleed to zero to avoid this.
- The number of sections is the number of parts in the Print context. Although this Template wizard can add multiple Print sections to the Print context, it will only add content to the first section.

On the next settings page (click **Next** to go there), you can type a subject, the sender's name and the sender's title. These will appear in the letter. You can also:

- Click the **Browse** button to select a signature image. This image will appear above the sender's name and title.
- Select Virtual Stationery: a PDF file with the letterhead stationery. Also see Media.

When you click **Finish**, the Wizard creates:

- A Print context with one section in it; see "Print context" on page 132 and "Print sections" on page 137.
- One empty Master Page. Master Pages are used for headers and footers, for images and other elements that have to appear on more than one page, and for special elements like tear-offs. See "Master Pages" on page 156.
- One Media. You can see this on the **Resources** pane: expand the **Media** folder. **Media 1** is the Virtual Stationery that you have selected in the Wizard. It is applied to all pages in the Print section, as can be seen in the Sheet Configuration dialog. (To open this dialog, expand the **Contexts** folder on the **Resources** pane; expand the **Print** folder and right-click "Section 1"; then select **Sheet Configuration**.) See "Media" on page 159.
- **Selectors** for variable data, for example: @Recipient@. You will want to replace these by the names of fields in your data. See "Variable Data" on page 351.

The Wizard opens the Print section. You can add text and other elements; see "Content elements" on page 191.

The formal letter template already has an address on it. The address lines are paragraphs, located in one cell in a table with the ID **address-block-table**. As the table has no borders, it is initially invisible. The address lines will stick to the bottom of that cell, even when the address has fewer lines. See "Styling and formatting" on page 294 to learn how to style elements.

> **Tip**
>
> Click the **Edges** button on the toolbar t emporarily adds a frame to certain elements on the Design tab. These will not .Print or output.

PDF-based Print template

> **Tip**
>
> The quickest way to create a Print template based on a PDF file is to **right-click the PDF** file in the Windows Explorer and select **Enhance with Connect**.

The PDF-based Print template wizard creates a document from an existing PDF file: a brochure, voucher, letter, etc. The PDF is used as the background image of the Print section (see "Using a PDF file or other image as background" on page 142). Variable and personalized

elements, like a reseller address, voucher codes and so on, can be added in front of it (see "Personalizing content" on page 338 and "Variable Data" on page 351).

By default, the PDF itself is added to the **Image** folder located in the **Resources** pane. Uncheck the option **Save with template** if the PDF should not be imported in the template. If it isn't saved with the template, the image remains external. Note that external images need to be available when the template is merged with a record set to generate output, and that their location should be accessible from the machine on which the template's output is produced. External images are updated (retrieved) at the time the output is generated.

After clicking **Next**, you can change the settings for the page. The initial page size and bleed area are taken from the selected PDF.

When you click **Finish**, the Wizard creates:

- A Print context with one section in it; see "Print context" on page 132 and "Print sections" on page 137. The selected PDF is used as the background of the Print section; see "Using a PDF file or other image as background" on page 142. For each page in the PDF one page is created in the Print section.
- One empty Master Page. Master Pages are used for headers, footers, images and other elements that have to appear on more than one page, and for special elements like tear-offs. See "Master Pages" on page 156.
- One empty Media. Media, also called Virtual Stationery, can be applied to all pages in the Print section. See "Media" on page 159.

Word-based Print template

The Word-based Print template wizard creates a document from an existing Microsoft Word file: a brochure, voucher, letter, mail merge document, etc. The images, lists, tables, and other content within the document are imported into Designer and used for a new template.

After clicking **Next**, you can change the settings for the page. The initial page size, margins, and orientation are taken from the original document settings, and you can adjust them on the dialog. You can also select the number of sections and a minimum number of pages. After the template is created, you can change the size of images and text boxes, and adjust the page size and margins.

When you click **Finish**, the Wizard creates:

- A Print context with the specified number of sections in it; see "Print context" on the facing page and "Print sections" on page 137. For each page in the document one page is created in the Print section.
- Image files for each image in the original file. As with other templates, you can place images on the master page if you want them to appear in every document in the same place.
- A separate word.css file in the Stylesheets. The css file is linked to the active section.
- A folder to contain the imported image files.

If the Word document contains mail merge fields, these mail merge fields (or markers) are converted into data fields.

- The brackets from the Word mail merge fields are converted to the @ character.
- The variable is wrapped with a span element.
- A user script is created for each data field.
- The mail merge fields are added to the Data Model of the OL Connect template.

See "Creating a Word-Based Print template with mail merge" on page 114 for details.

ERP templates

The ERP template wizard creates a business document. There is a collection of business documents that you can choose from: Sales Invoice, Purchase Order, Collection Letter, etc.. Currently all of these documents follow the corporate style designed by Microspective.

The first page of the wizard lets you select the page settings, see "Page settings: size, margins and bleed" on page 148. A few clarifications:

- Duplex means double-sided printing.
- The margins define where your text flow will go. The actual printable space on a page depends on your printer.
- The bleed is the printable space **around** a page. It can be used on some printers to ensure that no unprinted edges occur in the final trimmed document. Printers that can't print a bleed, will misinterpret this setting. Set the bleed to zero to avoid this.
- The number of sections is the number of parts in the Print context. Although this Template wizard can add multiple Print sections to the Print context, it will only add content to the first section.

On the next settings page (click **Next** to go there):

- Choose the desired type of business document from the **General** drop-down.
- Select a color for the colored parts of the document; see "Color Picker" on page 492.
- Enter your contact details.
- Click the **Browse** button to select a logo, or select to use a placeholder logo or no logo at all.
- Select a PDF file with the letterhead stationery. Also see "Media" on page 159.

> **Tip**
>
> Nice to know: your info and preferences are saved and will be reused the next time you create an ERP template.

When you click **Finish**, the Wizard creates:

- A Print context with one section in it; see "Print context" below and "Print sections" on page 137.
- One Master Page. Master Pages are used for headers and footers, for images and other elements that have to appear on more than one page, and for special elements like tear-offs. See "Master Pages" on page 156.
- One Media. You can see this on the **Resources** pane: expand the **Media** folder. **Media 1** is the Virtual Stationery that you have selected in the Wizard. It is applied to all pages in the Print section, as can be seen in the Sheet Configuration dialog. (To open this dialog, expand the **Contexts** folder on the **Resources** pane; expand the **Print** folder and right-click "Section 1"; then select **Sheet Configuration**.) See "Media" on page 159.
- **Selectors** for variable data, for example: @Name@, @Amount@. You will want to replace these by the names of actual fields in your data. See "Variable Data" on page 351.

The Wizard opens the Print section. You can add text and other elements; see "Content elements" on page 191. See "Styling and formatting" on page 294 to learn how to style elements..

Print context

The Print context is the folder in the Designer that can contain one or more Print templates.

Print templates (also called Print *sections*), are part of the Print context. They are meant to be printed directly to a printer or a printer stream/spool file, or to a PDF file (see "Generating Print output" on page 993).
The Print context can also be added to Email output as a PDF attachment; see "Generating Email output" on page 1017.
When generating output from the Print context, each of the Print sections is added to the output document, one after the other in sequence, for each record.

Creating the Print context

You can start creating a Print template with a Wizard (see "Creating a Print template with a Wizard" on page 126), or add the Print context to an existing template (see "Adding a context" on page 119).

> **Tip**
>
> Editing PDF files in the Designer is not possible, but when they're used as a section's background, you can add text and other elements, such as a barcode, to them.
> The quickest way to create a Print template based on a PDF file is to right-click the PDF file in the Windows Explorer and select **Enhance with Connect**. Alternatively, start creating a new Print template with a Wizard, using the PDF-based Print template (see "Creating a Print template with a Wizard" on page 126).
> To use a PDF file as background image for an existing section, see "Using a PDF file or other image as background" on page 142.

When a Print template is created, the following happens:

- The Print context is created and one **Print section** is added to it. You can see this on the **Resources** pane: expand the **Contexts** folder, and then expand the **Print** folder.
  The Print context can contain multiple sections: a covering letter and a policy, for example, or one section that is meant to be attached to an email as a PDF file and another one that is going to be printed out on paper. Only one Print section is added to it at the beginning, but you can add as many print sections as you need; see "Adding a Print section" on page 138.
  See "Print sections" on page 137 to learn how to fill a Print section.
- One **Master Page** is added to the template, as can be seen on the **Resources** pane, in the **Master Page** folder.

In Print sections, there are often elements that need to be repeated across pages, like headers, footers and logos. In addition, some elements should appear on each first page, or only on pages in between the first and the last page, or only on the last page. Examples are a different header on the first page, and a tear-off slip that should show up on the last page.

This is what Master Pages are used for. Master Pages can only be used in the Print context.
See "Master Pages" on page 156.
Initially, the (empty) master page that has been created with the Print context will be applied to all pages in the Print section, but more Master Pages can be added and applied to different pages.

- One **Media** is added to the template, as is visible on the **Resources** pane, in the **Media** folder. This folder can hold the company's stationery in the form of PDF files. When applied to a page in a Print section, Media can help prevent the contents of a Print section from colliding with the contents of the stationery. See "Media" on page 159 to learn how to add Media and, optionally, print them.
  Initially, the (empty) media that has been created with the Print context, is applied to all pages in the Print section. You can add more Media and apply them each to different pages.
- One **Stylesheet**, named context_print_styles.css, is added to the template, as you can see on the Resources pane, in the **Stylesheets** folder. This stylesheet is meant to be used for styles that are only applied to elements in the Print context. See also "Styling templates with CSS files" on page 296.

Print settings in the Print context and sections

The following settings in the Print context and Print sections have an impact on how the Print context is printed.

Arranging and selecting sections

The Print context can contain one or more Print sections. When generating output from the Print context, each of the Print sections is added to the output document, one after the other in sequence, for each record. The sections are added to the output in the order in which they appear on the **Resources** pane. This order can be changed; see "Print sections" on page 137.

It is also possible to exclude sections from the output, or to include a section only on a certain condition that depends on a value in the data; see "Conditional Print sections" on page 363. This can also be done using a Control Script; see "Control Scripts" on page 429.

Printing on both sides

To print a Print section on both sides of the paper, that Print section needs to have the Duplex printing option to be enabled; see "Enabling double-sided printing (Duplex, Mixplex)" on page 145. This setting can not be changed in a Job Creation Preset or an Output Creation Preset.

> **Note**
>
> Your printer must support duplex for this option to work.

Setting the binding style for the Print context

The Print context , as well as each of the Print sections, can have its own Finishing settings. In printing, Finishing is the way pages are bound together after they have been printed. Which binding styles can be applied depends on the type of printer that you are using.

To set the binding style of the Print **context**:

1. On the **Resources** pane, expand the **Contexts** folder; then right-click the **Print** context and select **Finishing**.
   Alternatively, select **Context > Finishing** on the main menu. This option is only available when editing a Print section in the Workspace.
2. Choose a Binding style and, if applicable, the number of holes. For an explanation of all Binding and Hole making options, see "Finishing options" on page 691.

To set the binding style of a Print **section**, see "Setting the binding style for a Print section" on page 145.

Overriding binding styles in a job creation preset

A *Job Creation Preset* can override the binding styles set for the Print sections and for the Print context as a whole. To bind output in another way than defined in the template's settings:

1. Create a Job Creation Preset that overrides the settings of one or more sections: select **File > Presets** and see "Job Creation Presets Wizard" on page 684 for more details.
2. Select that Job Creation Preset in the Print wizard; see "Generating Print output" on page 993.

Setting the bleed

The **bleed** is the printable space around a page. It can be used on some printers to ensure that no unprinted edges occur in the final trimmed document. The bleed is one of the settings for a section. See "Page settings: size, margins and bleed" on page 148.

Overprint and black overprint

Normally, when two colors overlap in Print output, the underlying color is not printed. It is "knocked out", for two reasons: firstly, the underlying color may affect the top color, especially if the top color is lighter than the underlying color. Secondly, not printing an underlying color, which is not visible anyway, will save ink or toner.
However, there are cases when underlying colors should not be knocked out:

- If the top color is a **special ink or toner**, such as varnish or UV, it should go over any other colors, as it is meant to be transparent and go over other content. In Connect you may enable overprint when you define a spot color; see "Defining colors, spot colors and tints" on page 328.
- If **small black text** is printed over a colored area, mis-registration may cause white areas to be visible around the text if the underlying color is knocked out.

The option to print small black text over other colors is referred to as **black overprint**. To enable black overprint for text smaller than a given size:

1. Right-click the **Print context** in the **Resources** pane and select **Color Output**.
2. In the **Text smaller than** field, enter a text size (in points, for example: 3pt).

> **Note**
>
> Black overprint only works with CMYK black.

Converting RGB black to CMYK black

In Print output, black is by default output as a CMYK color. RGB black is converted into CMYK black.
If that is not desirable for any reason, it is possible to prevent it:

1. Right-click the **Print context** in the **Resources** pane and select **Color Output**.
2. Enable the **Keep RGB black in output** option.

In Connect versions prior to 2018.2, RGB black was not automatically converted to CMYK black. Therefore, this option is by default enabled in templates made with an earlier version. In new templates, this option is disabled by default.

Print sections

Print templates (also called Print *sections*), are part of the Print context. They are meant to be printed directly to a printer or a printer stream/spool file, or to a PDF file (see "Generating Print output" on page 993).
The Print context can also be added to Email output as a PDF attachment; see "Generating Email output" on page 1017.
When generating output from the Print context, each of the Print sections is added to the output document, one after the other in sequence, for each record.

Pages

Unlike emails, Print sections can contain multiple *pages*. Pages are naturally limited by their size and margins. If the content of a section doesn't fit on one page, the overflow goes to the next page. This happens automatically, based on the section's page size and margins; see "Page settings: size, margins and bleed" on page 148.
The minimum number of pages can be set via the Print section properties; see "Print section properties" on page 542.

Although generally the same content elements can be used in all three contexts (see "Content elements" on page 191), the specific characteristics of pages make it possible to use special elements, such as page numbers; see "Page numbers " on page 150.

See "Pages" on page 147 for an overview of settings and elements that are specific for pages.

Using headers, footers, tear-offs and repeated elements

In Print sections, there are often elements that need to be repeated across pages, like headers, footers and logos. In addition, some elements should appear on each first page, or only on pages in between the first and the last page, or only on the last page. Examples are a different header on the first page, and a tear-off slip that should show up on the last page.

This is what Master Pages are used for. Master Pages can only be used in the Print context.

See "Master Pages" on page 156 for an explanation of how to fill them and how to apply them to different pages.

Using stationery (Media)

When the output of a Print context is meant to be printed on paper that already has graphical and text elements on it (called stationery, or preprinted sheets), you can add a copy of this media, in the form of a PDF file, to the Media folder.

Media can be applied to pages in a Print section, to make them appear as a background to those pages. This ensures that elements added to the Print context will correspond to their correct location on the preprinted media.

> **Note**
>
> When both Media and a Master Page are used on a certain page, they will both be displayed on the Preview tab of the workspace, the Master Page being 'in front' of the Media and the Print section on top. To open the Preview tab, click it at the bottom of the Workspace or select **View > Preview View** on the menu.

See "Media" on page 159 for a further explanation about how to add Media and how to apply them to different pages.

Note: The Media will not be printed, unless this is specifically requested through the printer settings; see "Generating Print output" on page 993.

Copy Fit

Copy Fit is a feature to automatically adjust the font size of text to make it fit the available space. It could be used for the name of a person on a greeting card, for instance, or for the name of a product on a shelf talker. This feature is only available with Box and Div elements in Print sections.

For more information about this feature see "Copy Fit" on page 311.

Adding a Print section

The Print context can contain multiple sections: a covering letter and a policy, for example, or one section that is meant to be attached to an email as a PDF file and another one that is meant

to be printed out on paper. When a Print template is created (see "Creating a Print template with a Wizard" on page 126 and "Print context" on page 132), only one Print section is added to it, but you can add as many print sections as you need.

To add a section to a context:

- On the **Resources** pane, expand the **Contexts** folder, right-click the **Print** context , and then click **New section**.

Note that the new section automatically gets the same properties as the first section.

The first Master Page (see "Master Pages" on page 156) and Media (see "Media" on page 159) will automatically be applied to all pages in the new section, but this can be changed, see "Applying a Master Page to a page in a Print section" on page 158 and "Applying Media to a page in a Print section" on page 163.

Note that Print **sections** always start on a front page. To avoid empty pages between sections, consider combining them into one section, and use page breaks to push content to the next page (see "Page breaks" on page 154).

Tip

Editing PDF files in the Designer is not possible, but when they're used as a section's background, you can add text and other elements, such as a barcode, to them.
The quickest way to create a Print template based on a PDF file is to right-click the PDF file in the Windows Explorer and select **Enhance with Connect**. Alternatively, start creating a new Print template with a Wizard, using the PDF-based Print template (see "Creating a Print template with a Wizard" on page 126).
To use a PDF file as background image for an existing section, see "Using a PDF file or other image as background" on page 142.

Note

Via a Control Script, sections can be added to a Print context dynamically; see "Dynamically adding sections (cloning)" on page 440.

Tip

If you need a whole Print section to be visible in the output only under certain conditions, consider using the Conditional Print Section script wizard; see "Conditional Print sections" on page 363.

You can use the Conditional Content script wizard to hide parts of the content of a section; see "Showing content conditionally" on page 358.

Importing a Print section

To import a section from another template, click **File > Import Resources...** in the menu. See: "Import Resources dialog" on page 502.
Remember also to add or import any related source files, such as images.

Note that when the imported Print section *replaces* a Print section in your template, its context's Color Output and Finishing settings get imported as well. (See "Print settings in the Print context and sections" on page 134.)

Deleting a Print section

To delete a Print section:

- On the **Resources** pane, expand the **Contexts** folder, expand the **Print** context, right-click the name of the section, and then click **Delete**.

Warning

If you don't have a backup of the template, the only way to recover a deleted section, is to click **Undo** on the **Edit** menu, until the deleted section is restored. After closing and reopening the template it is no longer possible to restore the deleted context this way.
In the Saving Preferences you can set whether a backup file should be created when you save the template; see "Save preferences" on page 391.

Arranging Print sections

When generating output from the Print context, each of the Print sections is added to the output document, one after the other in sequence, for each record. The sections are added to the

output in the order in which they appear on the **Resources** pane, so changing the order of the sections in the Print context changes the order in which they are outputted to the final document.

To rearrange sections in a context:

- On the **Resources** pane, expand the **Print** context and drag and drop sections to change the order they are in.
- Alternatively, on the **Resources** pane, right-click a section in the **Print** context and click **Arrange**. In the Arrange Sections dialog you can change the order of the sections by clicking the name of a section and moving it using the **Up** and **Down** buttons.

Styling and formatting a Print section

The contents of a Print section can be formatted directly, or styled with Cascading Style Sheets (CSS). See "Styling and formatting" on page 294.

In order for a style sheet to be applied to a specific section, it needs to be included in that section. There are two ways to do this.

Drag & drop a style sheet

1. Click and hold the mouse button on the style sheet on the **Resources** pane.
2. Move the mouse cursor within the **Resources** pane to the section to which the style sheet should be applied.
3. Release the mouse button.

Using the Includes dialog

1. On the Resources pane, right-click the section, then click **Includes**.
2. From the **File types** dropdown, select **Stylesheets**.
3. Choose which CSS files should be applied to this section. The available files are listed at the left. Use the arrow buttons to move the files that should be included to the list at the right.
4. You can also change the order in which the CSS files are read: click one of the included CSS files and use the **Up** and **Down** buttons. Note that moving a style sheet up in the list gives it **less** weight. In case of conflicting rules, style sheets read later will override previous ones.

Note

Style sheets are applied in the order in which they are included in a section. The styles in each following style sheet add up to the styles found in previously read style sheets. When style sheets have a conflicting rule for the same element, class or ID, the **last** style sheet 'wins' and overrides the rule found in the previous style sheet.

Note

Style sheets that are linked to (i.e. included in) a section show a chain icon in the Resources pane (see "Resources pane" on page 583).

Using a PDF file or other image as background

In the Print context, a PDF file can be used as a section's background. It is different from the Media in that the section considers the PDF to be content, so the number of pages in the section will be the same as the number of pages taken from the PDF file.

With this feature it is possible to create a Print template from an arbitrary PDF file. Of course, the PDF file itself can't be edited in a Designer template, but when it is used as a section's background, text and other elements, such as a barcode, can be added to it.

Encrypted PDF files are **not supported** in *PDF pass-through* mode.

To use a PDF file as background image:

1. On the **Resources** pane, expand the **Print** context, right-click the print section and click **Background**.
2. Click the downward pointing arrow after **Image** and select **From PDF resource**.
3. For a PDF resource, you have to specify the **path**. Clicking the **Select Image** button opens the Select Image dialog (see "Select Image dialog" on page 545).
4. - Click **Resources**, **Disk** or **Url**, depending on where the image is located.
   - **Resources** lists the images that are present in the **Images** folder on the **Resources** pane.

- **Disk** lists image files that reside in a folder on a hard drive that is accessible from your computer. Click the **Browse** button to select a folder (or an image in a folder).
  As an alternative it is possible to enter the path manually. You can give a local path (e.g. C:\Images\Test.jpg) or use the "file" protocol. The complete syntax of a fully qualified URL with the "file" protocol is: `file://<host>/<path>`.
  Note: if the host is "localhost", it can be omitted, resulting in `file:///<path>`, for example: `file:///c:/resources/images/image.jpg`.
- **Url** lists image files from a specific web address. Select the protocol (**http** or **https**), and then enter a web address (for example, http://www.mysite.com/images/image.jpg).

> **Note**
>
> If a URL doesn't have a file extension, and the option Save with template is **not** selected, the Select Image dialog automatically adds the `filetype` parameter with the file extension as its value (for example: `?filetype=pdf` (if it is the first parameter) or `&filetype=pdf`).
> The `filetype`, `page` and `nopreview` parameters are not sent to the host; they are used internally. Therefore, URLs that rely on one of these parameters cannot be used.

- With an external image, you can check the option **Save with template**. If this option is checked, the file will be inserted in the **Images** folder on the **Resources** pane at the top left.
  If it isn't saved with the template, the image remains external. Note that external images need to be available when the template is merged with a record set to generate output, and that their location should be accessible from the machine on which the template's output is produced. External images are updated (retrieved) at the time the output is generated.

5. Select the PDF's **position**:
   - **Fit to page** stretches the PDF to fit the page size.
   - **Centered** centers the PDF on the page, vertically and horizontally.
   - **Absolute** places the PDF at a specific location on the page. Use the **Top** field to specify the distance between the top side of the page and the top side of the PDF,

and the **Left** field to specify the distance between the left side of the page and the left side of the PDF.
The Top and Left offset can be specified in the usual units of measurement or as a percentage of the page (for example: a Left value of 25% means it will be placed at 25% of the page width).

6. Set the **scale** of the width (x) and height (y) of the image as a percentage of the original image. To avoid stretching, check the option **Keep aspect ratio** and set the scale of the width.
7. Click one of the options next to **Rotation** to rotate the image.
8. Optionally, if the PDF has more than one page, you can set the range of **pages** that should be used.

> **Note**
>
> The number of pages in the Print section is automatically adjusted to the number of pages in the PDF file that are being used as the section's background image.

9. Finally, click **OK**.

> **Tip**
>
> An alternative to using a PDF as background inside the template is to layer the template (i.e. the PDF output of a Print section) over the background PDF via a Script task in a Workflow process. This is called 'stamping'. In the unusual case where extracting text from the PDF that is the output of a Print section with a PDF background doesn't work, it is recommended to use this method. For more information, see this how-to: Stamping one PDF file on another.

Dynamic backgrounds

To make the background change based on the value of a data field, you may use the Dynamic Background Script Wizard; see "Dynamic Print section backgrounds" on page 368.

Alternatively you could write your own Control Script to set the background; see "Control Script: Setting a Print section's background" on page 438.

The settings in a script take precedence over the settings made in the Print Section Properties dialog.

Setting the binding style for a Print section

In printing, Finishing is the binding style, or the way pages are bound together. Each Print section can have its own Finishing settings, as well as the Print context as a whole; see "Setting the binding style for the Print context" on page 135.

To set the binding style of a Print **section**:

1. On the **Resources** pane, expand the **Contexts** folder, expand the **Print** context and right-click the Print section.
2. Click **Finishing**.
3. Choose a Binding style and, if applicable, the number of holes.

Overriding binding styles in a job creation preset

A *Job Creation Preset* can override the binding styles set for the Print sections and for the Print context as a whole. To bind output in another way than defined in the template's settings:

1. Create a Job Creation Preset that overrides the settings of one or more sections: select **File > Presets** and see "Job Creation Presets Wizard" on page 684 for more details.
2. Select that Job Creation Preset in the Print wizard; see "Generating Print output" on page 993.

Enabling double-sided printing (Duplex, Mixplex)

To print a Print section on both sides of the paper, that Print section needs to have the Duplex printing option to be enabled. This is an option in the Sheet Configuration dialog. (See "Sheet Configuration dialog" on page 546.)

> **Note**
>
> Your printer must support Duplex for this option to work.

To enable Duplex or Mixplex printing:

1. On the **Resources** pane, expand the **Print** context, right-click the print section and click **Sheet configuration**.
2. Check **Duplex** to enable content to be printed on the back of each sheet.
3. When Duplex printing is enabled, further options become available.
    - Check **Omit empty back side for Last or Single sheet** to reset a page to Simplex if it has an empty back side. **This changes the Duplex job into a Mixplex job.** Thus changing a Duplex job into a **Mixplex** job may reduce volume printing costs as omitted back sides aren't included in the number of printed pages. On the other hand, depending on the printer type it may reduce the print speed as well.
      See also: "The consequences of empty back sides for printing and page numbering" below.

      Note

      Use a **print format** that supports Duplex/Simplex switching to get a proper Mixplex job. PDF is not suitable as it has no device control.

      Print **sections** always start on a front page. To avoid empty pages between sections, consider combining them into one section, and use page breaks to push content to the next page (see "Page breaks" on page 154).

    - Check **Tumble** to duplex pages as in a calendar.
    - Check **Facing pages** to have the side margins switched alternately, so that after printing and binding the pages, they look like in a magazine or book. See "Pages" on the next page to find out how to set a left and right margin on a page.

Note

Master Pages, Media and Duplex printing options can also be set in a Control Script (see "Control Scripts" on page 429 and "Control Script API" on page 931). This is especially useful when you need identical sections with different settings.

The consequences of empty back sides for printing and page numbering

In a Duplex job, the last page of a section may be empty since **each new section starts on a new sheet**. You may wonder what this means for the number of 'print clicks' and for the page

numbering.

Note that an empty page is defined as a page that has no content **and** no Master Page. To suppress the Master Page on otherwise empty back sides, check the option **Omit Master Page Back in case of an empty back page** for last and single sheets, in the Sheet Configuration dialog. (See "Sheet Configuration dialog" on page 546.)

As of version 2020.2, a page that only has a DataMapper PDF background is no longer seen as empty. This may affect the output of templates created with previous versions.

Print clicks

If a page is empty, but still sent to a printer, it may be counted as a 'click' on the printer.
To avoid this, you could check the **Omit empty back side for Last or Single sheet** option in the Duplex printing settings. This resets a page to Simplex if it has an empty back side. The resulting print job can then be "mixed plex"; in that contains both simplex and duplex pages. The omitted back side isn't sent to the printer, so it doesn't count as a print click.
If a page is empty, but not omitted (the Omit empty back side option is not checked) it will still be sent to the printer and may count as a print click.

Page numbers

An empty back side that is omitted from the output does not count in the page numbers either.

If a back side is empty, but not omitted (the Omit empty back side for Last or Single sheet option is not checked) it will be skipped from the page count *unless* the page numbers continue on the next section (see "Configuring page numbers" on page 152).

Pages

Unlike emails, Print sections can contain multiple *pages*. Pages are naturally limited by their size and margins. If the content of a section doesn't fit on one page, the overflow goes to the next page. This happens automatically, based on the section's page size and margins; see "Page settings: size, margins and bleed" on the facing page.
The minimum number of pages can be set via the Print section properties; see "Print section properties" on page 542.

Although generally the same content elements can be used in all three contexts (see "Content elements" on page 191), the specific characteristics of pages make it possible to use special elements, such as page numbers; see "Page numbers " on page 150.

The widow/orphan setting lets you control how many lines of a paragraph stick together, when content has to move to another page; see "Preventing widows and orphans" on page 152. You can also avoid or force a page break before or after an entire element, see "Page breaks" on page 154.

Each page in a print section has a natural position: it is the first page, the last page, a 'middle' page (a page between the first and the last page) or a single page. For each of those positions, a different Master Page and Media can be set. A Master Page functions as a page's background, with for example a header and footer. A Media represents preprinted paper that a page can be printed on. See "Master Pages" on page 156 and "Media" on page 159.

Page specific content elements

The specific characteristics of pages make it possible to use these special elements:

- **Page numbers** can only be used in a Print context. See "Page numbers " on page 150 to learn how to add and change them.
- Conditional content and dynamic tables, when used in a Print section, may or may not leave an empty space at the bottom of the last page. To fill that space, if there is any, an image or advert can be used as a **whitespace element**; see "Whitespace elements: using optional space at the end of the last page" on the next page.
- Dynamic Tables can be used in all contexts, but **transport lines** are only useful in a Print context; see Dynamic Table.

Positioning and aligning elements

Sometimes, in a Print template, you don't want content to move up or down with the text flow. To prevent that, put that content in a Positioned Box. See "Content elements" on page 191.

When it comes to positioning elements on a page, Guides can be useful, as well as Tables. See "How to position elements" on page 312.

Page settings: size, margins and bleed

On paper, whether it is real or virtual, content is naturally limited by the page size and margins.

These, as well as the bleed, are set per Print section, as follows:

- On the **Resources** pane, right-click a section in the **Print** context and click **Properties**.

For the **page size**, click the drop-down to select a page size from a list of common paper sizes. Changing the width or height automatically sets the page size to Custom.

Margins define where your text flow will go. Static elements can go everywhere on a page, that is to say, within the printable space on a page that depends on the printer.

The **bleed** is the printable space around a page. It can be used on some printers to ensure that no unprinted edges occur in the final trimmed document. Note: Printers that can't print a bleed, will misinterpret this setting. Set the bleed to zero to avoid this.

> **Tip**
>
> By default, measurements settings are in inches (in). You could also type measures in centimeters (add 'cm' to the measurement, for example: 20cm) or in millimeters (for example: 150mm).
> To change the default unit for measurement settings to centimeters or millimeters: on the menu, select **Window > Preferences > Print > Measurements**.

Whitespace elements: using optional space at the end of the last page

Print sections with conditional content and dynamic tables (see "Personalizing content" on page 338) can have a variable amount of space at the bottom of the last page. It is useful to fill the empty space at the bottom with transpromotional material, but of course you don't want extra pages created just for promotional data. 'Whitespace elements' are elements that will only appear on the page if there is enough space for them.

To convert an element into a whitespace element:

1. Import the promotional image or snippet; see "Images" on page 271 and "Snippets" on page 285.

2. Insert the promotional image or snippet in the content.

> **Note**
>
> - Only a top-level element (for example, a paragraph that is not inside a table or div) can function as a whitespace element.
> - Do not place the promotional image or snippet inside an absolute positioned box. Whitespacing only works for elements that are part of the text flow, not for absolute-positioned boxes.

3. Select the image or the element that holds the promotional content: click it, or use the breadcrumbs, or select it on the **Outline** tab; see "Selecting an element" on page 196.
4. On the **Attributes** pane, check the option **Whitespace element**.
5. (Optional.) Add extra space at the top of the element: on the menu **Format**, click the option relevant to the selected element (Image for an image, Paragraph for a paragraph, etc.) and adjust the spacing (padding and/or margins).
   Do not add an empty paragraph to provide space between the whitespace element and the variable content. The extra paragraph would be considered content and could end up on a separate page, together with the whitespace element.

Page numbers

Inserting page numbers

Page numbers can be added to a Print section, but they are usually added to a Master Page, because headers and footers are designed on Master Pages; see also: "Master Pages" on page 156.

To insert a page number, select **Insert > Special character > Markers** on the menu, and then click one of the options to decide with what kind of page number the marker will be replaced:

- **Page number**: The current page number in the document. If a page is empty or does not display a page number, it is still added to the page count.
- **Page count**: The total number of pages in the document, including pages with no contents or without a page number.
- **Content page number**: The current page number in the document, counting only pages with contents that are supplied by the Print section. A page that has a Master Page (as set

in the Sheet Configuration dialog, see "Applying a Master Page to a page in a Print section" on page 158) but no contents, is not included in the Content page count.

- **Content page count**: This is the total number of pages in the current document that have contents, supplied by the Print section. A page that has a Master Page but no contents, is not included in the Content page count.
- **Sheet number**: The current sheet number in the document. A sheet is a physical piece of paper, with two sides (or pages). This is equivalent to half the page number, for example if there are 10 pages, there will be 5 sheets.
- **Sheet count**: This marker is replaced by the total number of sheets in the document, whether or not they have contents.

Note

When a marker is inserted, a class is added to the element in which the marker is inserted. Do not delete that class. It enables the software to quickly find and replace the marker when generating output. The respective classes are: `pagenumber`, `pagecount`, `contentpagenumber`, `contentpagecount`, `sheetnumber`, and `sheetcount`.

Tip

Instead of page numbers, you might want to display the current **record index** and/or the total number of records in the record set, in the document. There is a How-to that explains how to do that: How to get the record index and count.

Creating a table of contents

A table of contents can only be created in a script.
If you are looking to create a short, simple table of contents in **one section**, you could add a Standard Script that uses the `pageRef()` function. For an example, see "pageref()" on page 919.
For a **multi-page**, **cross-section** table of contents you must use a Post Pagination Script; see "Creating a Table Of Contents" on page 447.

The basics of script-writing in the Designer are explained in the following topic: "Writing your own scripts" on page 393.

Configuring page numbers

By default the page numbers are Arabic numerals (1, 2, 3, etc.) without leading zeros nor prefix, and page numbering starts with page 1 for each section. But this can be changed. To do that:

1. On the **Resources** pane, right-click a section in the **Print** context and click **Numbering**.
2. Uncheck **Restart Numbering** if you want the page numbers to get consecutive page numbers, instead of restarting the page numbering with this section.

> **Note**
>
> Even if a section is disabled, so it doesn't produce any output, this setting is still taken into account for the other sections. This means that if Restart Numbering is checked on a disabled section, the page numbering will be restarted on the next section.
> Disabling a section can only be done in a Control Script (see "Control Scripts" on page 429). Control Scripts can also change where page numbers restart.

3. Use the **Format** drop-down to select uppercase or lowercase letters or Roman numerals instead of Arabic numerals.
4. In **Leading Zeros**, type zeros to indicate how many digits the page numbers should have. Any page number that has fewer digits will be preceded by leading zeros.
5. Type the **Number prefix**. Optionally, check Add Prefix to Page Counts, to add the prefix to the total number of pages, too.
6. Close the dialog.

Preventing widows and orphans

Widows and orphans are lines at the beginning or at the end of a paragraph respectively, dangling at the bottom or at the top of a page, separated from the rest of the paragraph.
By default, to prevent orphans and widows, lines are moved to the next page as soon as two lines get separated from the rest of the paragraph. The same applies to list items (in unordered, numbered and description lists).
The number of lines that should be considered a widow or orphan can be changed for the entire Print context, and per paragraph and in tables.

Note

Widows and orphans are ignored if the **page-break-inside** property of the paragraph is set to **avoid**; see "Preventing a page break" on page 155.

In the entire Print context

To prevent widows and orphans in the entire Print context:

1. On the menu, select **Edit > Stylesheets**.
2. Select the **Print** context.
3. Click **New** (or, when there are already CSS rules for paragraphs, click the selector **p** and click **Edit**).
4. Click **Format**.
5. After **Widows and Orphans**, type the minimum number of lines that should be kept together.

Alternatively, manually set the set the **widows** and **orphans** properties in a style sheet:

1. Open the style sheet for the Print context: on the **Resources** pane, expand the **Styles** folder and double-click context_print_styles.css.
2. Add a CSS rule, like the following:
   ```
   p { widows: 4; orphans: 3 }
   ```

Per paragraph

To change the widow or orphan setting for one paragraph only:

1. Open the Formatting dialog. To do this, you can:
   - Select the paragraph using the breadcrumbs or the **Outline** pane (next to the **Resources** pane) and then select **Format > Paragraph** in the menu.
   - Right-click the paragraph and select **Paragraph...** from the contextual menu.
2. After **Widows** and **Orphans**, type the minimum number of lines that should be kept together.

In tables

The CSS properties **widows** and **orphans** can be used in tables to prevent a number of rows from being separated from the rest of the table.

A Standard Table doesn't flow over multiple pages by default. Splitting a Standard Table over multiple pages requires setting the Connect-specific `data-breakable` attribute on all of its rows. You can either open the Source tab, or write a script to replace each `<tr>` with `<tr data-breakable="">`. Note that the effect will only be visible in Preview mode.

To set the number of widows and orphans for a table:

1. Open the Formatting dialog. To do this, you can:
    - Select the table using the breadcrumbs or the **Outline** pane (next to the **Resources** pane) and then select **Format > Table** in the menu.
    - Right-click the paragraph and select **Table...** from the contextual menu.
2. After **Widows** and **Orphans**, type the minimum number of table rows that should be kept together.

Page breaks

A page break occurs automatically when the contents of a section don't fit on one page.

> **Note**
>
> Improved page breaking in Connect 2019.1 might impact upon templates made with earlier versions. See "Known Issues" on page 90.

Inserting a page break

To insert a page break before or after a certain element, set the `page-break-before` property or the `page-break-after` property of that element (a paragraph for example; see also "Styling text and paragraphs" on page 308):

1. Select the element (see "Selecting an element" on page 196).
2. On the **Format** menu select the respective element to open the Formatting dialog.

3. In the Breaks group, set the **before** or **after** property.
    - **Before**: Sets whether a page break should occur **before** the element. This is equivalent to the **page-break-before** property in CSS; see CSS page-break-before property for an explanation of the available options.
    - **After**: Sets whether a page break should occur **after** the element. Equivalent to the page-break-after property in CSS; see CSS page-break-after property for an explanation of the available options.

Click the button **Advanced** to add CSS properties and values to the inline style tag directly. Alternatively you could set this property on the Source tab in the HTML (for example: `<h1 style="page-break-before: always;">`), or add a rule to the style sheet; see "Styling your templates with CSS files" on page 300.

> **Note**
>
> You cannot use these properties on an empty or on absolute-positioned elements.

Preventing a page break

To prevent a page break inside a certain element, set the **page-break-inside** property of that element to **avoid**:

- Select the element (see "Selecting an element" on page 196).
- On the **Format** menu, select the respective element to open the Formatting dialog.
- In the **Breaks** group, set the **inside** property to **avoid**, to prevent a page break inside the element. For an explanation of all available options of the **page-break-inside** property in CSS, see CSS page-break-inside property.

Alternatively you could set this property on the Source tab in the HTML (for example: `<ul style="page-break-inside: avoid;">`), or add a rule to the style sheet; see "Styling your templates with CSS files" on page 300.

Adding blank pages to a section

How to add a blank page to a section is described in a how-to: Create blank page on field value.

Master Pages

In Print sections, there are often elements that need to be repeated across pages, like headers, footers and logos. In addition, some elements should appear only on specific pages, such as only the first page, or the last page, or only on pages in-between. Examples are a different header on the first page, and a tear-off slip that shows up on the last page.

This is what Master Pages are used for. Master Pages can only be used in the Print context (see "Print context" on page 132).

Master Pages resemble Print sections, and they are edited in much the same way (see "Editing a Master Page" on the next page) but they contain a single page and do not have any text flow. Only one Master Page can be applied per page in printed output. Then a Print template is created, one master page is added to it automatically. You can add more Master Pages; see "Adding a Master Page" below. Initially, the original Master Page will be applied to all pages, but different Master Pages can be applied to different pages; see "Applying a Master Page to a page in a Print section" on page 158.

Examples

There are a few How-tos that demonstrate the use of Master Pages:

- Showing a Terms and Conditions on the back of the first page only.
- A tear-off section on the first page of an invoice.
- Tips and tricks for Media and Master Pages.

Adding a Master Page

When a Print template is created, one master page is added to it automatically. Adding more Master Pages can be done as follows:

- On the **Resources** pane, right-click the **Master pages** folder and click **New Master Page**.
- Type a name for the master page.
- Optionally, set the margin for the header and footer. See "Adding a header and footer" on the next page.
- Click **OK**.

Initially, the master page that has been created together with the Print context will be applied to all pages in the Print section. After adding more Master Pages, different Master Pages can be

applied to different pages; see "Applying a Master Page to a page in a Print section" on the facing page.

Importing a Master Page

To import one or more Master Pages from another template, click **File > Import Resources...** in the menu. See: "Import Resources dialog" on page 502.

Editing a Master Page

Master Pages are edited just like sections, in the workspace. To open a Master Page, expand the **Master pages** folder on the **Resources** pane, and double-click the Master Page to open it. The drop-downs at the top of the Workspace let you select a Section and a Media (front or back) that will serve as a background to your Master Page design.

A Master Page can contain text, images and other elements (see "Content elements" on page 191), including variable data and dynamic images (see "Personalizing content" on page 338). All elements on a Master Page should have an absolute position or be inside an element that has an absolute position. It is good practice to position elements on a Master Page by placing them in a Positioned Box (see "Content elements" on page 191).

Keep in mind that a Master Page always remains a single page. Its content cannot overflow to a next page. Content that doesn't fit, will not be displayed.

> **Note**
>
> Editing the Master Page is optional. One Master Page must always exist in a Print template, but if you don't need it, you can leave it empty.

Adding a header and footer

Headers and footers are not designed as part of the contents of a Print section, but as part of a Master Page, which is then applied to a page in a print section.

To create a header and footer:

1. First insert elements that form the header or footer, such as the company logo and address, on the Master Page; see "Editing a Master Page" above.

2. Next, define the margins for the header and footer. The margins for a header and footer are set in the Master Page properties. This does not change the content placement within the Master Page itself; in Master Pages, elements can go everywhere on the page. Instead, the header and footer of the Master Page limit the text flow on pages in the Print sections to which this Master Page is applied. Pages in a Print section that use this Master Page cannot display content in the space that is reserved by the Master Page for the header and footer, so that content in the Print section does not collide with the content of the header and footer. To set a margin for the header and/or footer:
    1. On the **Resources** pane, expand the **Master pages** folder, right-click the master page, and click **Properties**.
    2. Fill out the height of the header and/or the footer. The contents of a print section will not appear in the space reserved for the header and/or footer on the corresponding master page.
3. Finally, apply the master page to a specific page in a print section. See "Applying a Master Page to a page in a Print section" below.

Applying a Master Page to a page in a Print section

Every sheet in a Print section has a natural position: it can be the **first**, the **last**, one of the sheets in between ('**middle**'), or a **single** sheet. For each of these positions, you can set a different Master Page and Media (see "Media" on the next page). It can even have two master pages, if printing is done on both sides (called duplex printing).

To apply Master Pages to specific page positions in a Print section:

1. On the **Resources** pane, expand the **Print** context; right-click the Print section, and click **Sheet configuration**.
2. Optionally, check **Duplex** to enable content to be printed on the back of each sheet. Your printer must support duplex for this option to work. If Duplex is enabled, you can also check **Tumble** to duplex pages as in a calendar, and **Facing pages** to have the margins of the section switch alternately, so that pages are printed as if in a magazine or book.
3. If the option **Same for all positions** is checked, the same Master Page will be applied to every page in the print section (and to both the front and the back side of the page if duplex printing is enabled). Uncheck this option.
4. Decide which Master Page should be linked to which sheet (position): click the downward pointing arrow after **Master Page Front** and select a Master Page.
   If Duplex is enabled, you can also select a Master Page for the back of the sheet and consequently, check **Omit Master Page Back in case of an empty back page** to omit

the specified Master Page on the last backside of a section if that page is empty. That page will then also be skipped from the page count *unless* the page numbers continue on the next section (see "Configuring page numbers" on page 152).
Note that if the **Omit empty back side for Last or Single sheet** option (see "General options" on page 547) is checked as well, the empty backside will not appear in the output at all and will not be counted in any case.

5. Optionally, decide which Media should be linked to each sheet.
6. If output documents can be so long that they cannot fit in one envelope, you may check the **Repeat sheet configuration** option to have the sheet configuration repeat every *n* number of pages.
7. Click OK to save the settings and close the dialog.

> **Note**
>
> Master Pages, Media and Duplex printing options can also be set in a Control Script (see "Control Scripts" on page 429 and "Control Script API" on page 931). This is especially useful when you need identical sections with different settings.

Deleting a Master Page

To delete a Master Page, expand the **Master pages** folder on the **Resources** pane, right-click the master page, and click **Delete**.

Note that one Master Page as well as one Media must always exist in a Print template. Just leave it empty if you don't need it.

Media

When the output of a Print context is meant to be printed on paper that already has graphical and text elements on it (called stationery, or preprinted sheets), you can add a copy of this media, in the form of a PDF file, to the Media folder.

Media can be applied to pages in a Print section, to make them appear as a background to those pages. This ensures that elements added to the Print context will correspond to their correct location on the preprinted media.

For further explanation about how to apply Media to different pages, see "Applying Media to a page in a Print section" on page 163.

Media will not be printed, unless you want them to; see below.

Per Media, a front and back can be specified and you can specify on what kind of paper the output is meant to be printed on. This includes paper weight, quality, coating and finishing; see "Setting Media properties" below.

Adding Media

To add a Media, right-click the **Media** folder on the **Resources** pane and select **New Media**.

The new Media is of course empty. You can specify two PDF files for the Media: one for the front, and, optionally, another for the back.

Importing Media

To import Media from another template, click **File > Import Resources...** in the menu. See: "Import Resources dialog" on page 502.

Specifying and positioning Media

Specifying a PDF for the front: the fast way

To quickly select a PDF file for the front of a Media, drag the PDF file from the Windows Explorer to one of the Media. The Select Image dialog opens; select an image and check the option **Save with template** if you want to insert the image into the **Images** folder on the **Resources** pane. (For PDF files selected by URL this option is always checked.)
Alternatively you could first import the PDF file to the **Images** folder on the **Resources** pane (using drag & drop) and drag it from there on one of the Media in the **Media** folder.
Either way, you cannot set any options.

To be able to specify a PDF file for both the front and the back of the Media, and to specify a position for the Media's PDF files, you have to edit the properties of the Media.

Setting Media properties

Media have a number of properties that you can set. You can change the Media's page size and margins (as long as it isn't applied to a section), you can specify a PDF file (or any other

type of image file) for both the front and the back of the Media, and you can determine how the virtual stationery should be positioned on the page. This is done as follows:

1. On the **Resources** pane, expand the **Contexts** folder, expand the **Media** folder, right-click the Media and click **Properties**.
2. Now you can change the name and page size of the Media. Note that it isn't possible to change the page size once the Media is applied to a section. Media can only be applied to sections that have the same size.
3. On the **Virtual Stationery** tab, you can click the **Select Image** button to select a PDF image file.

> **Note**
>
> Encrypted PDF files are **not supported** in *PDF pass-through* mode. See "PDF Options" on page 712.

- Click **Resources**, **Disk** or **Url**, depending on where the image is located.
  - **Resources** lists the images that are present in the **Images** folder on the **Resources** pane.
  - **Disk** lists image files that reside in a folder on a hard drive that is accessible from your computer. Click the **Browse** button to select a folder (or an image in a folder).
    As an alternative it is possible to enter the path manually. You can give a local path (e.g. C:\Images\Test.jpg) or use the "file" protocol. The complete syntax of a fully qualified URL with the "file" protocol is: `file://<host>/<path>`.
    Note: if the host is "localhost", it can be omitted, resulting in `file:///<path>`, for example: `file:///c:/resources/images/image.jpg`.
  - **Url** lists image files from a specific web address. Select the protocol (**http** or **https**), and then enter a web address (for example, http://www.mysite.com/images/image.jpg).

> **Note**
>
> If a URL doesn't have a file extension, and the option Save with template is

not selected, the Select Image dialog automatically adds the `filetype` parameter with the file extension as its value (for example: `?filetype=pdf` (if it is the first parameter) or `&filetype=pdf`).
The `filetype`, `page` and `nopreview` parameters are not sent to the host; they are used internally. Therefore, URLs that rely on one of these parameters cannot be used.

- With an external image, you can check the option **Save with template**. If this option is checked, the file will be inserted in the **Images** folder on the **Resources** pane at the top left.
  If it isn't saved with the template, the image remains external. Note that external images need to be available when the template is merged with a record set to generate output, and that their location should be accessible from the machine on which the template's output is produced. External images are updated (retrieved) at the time the output is generated.

4. If the PDF file consists of more than one page, select the desired page.

Note

The number of pages in a PDF file can not be determined via the HTTP and HTTPS protocols. If you wish to use a page other than page 1 in a remote PDF, check the option **Save with template**; then click OK and reopen the dialog. Next, on the Resources tab, select the PDF, and select a page.

5. Click **Finish**.
6. For each of the PDF files, select a **position**:
   - **Fit to page** stretches the PDF to fit the page size.
   - **Centered** centers the PDF on the page, vertically and horizontally.
   - **Absolute** places the PDF at a specific location on the page. Use the **Top** field to specify the distance between the top side of the page and the top side of the PDF, and the **Left** field to specify the distance between the left side of the page and the left side of the PDF.
7. Finally, click **OK**.

Setting the paper's characteristics

To set a Media's paper characteristics:

1. On the **Resources** pane, expand the **Contexts** folder, expand the **Media** folder, and right-click the Media. Click **Characteristics**.
2. Specify the paper's characteristics:
   - **Media Type**: The type of paper, such as Plain, Continuous, Envelope, Labels, Stationery, etc.
   - **Weight**: The intended weight of the media in grammage (g/m2).
   - **Front Coating**: The pre-process coating applied to the front surface of the media, such as Glossy, High Gloss, Matte, Satin, etc.
   - **Back Coating**: The pre-process coating applied to the back surface of the media.
   - **Texture**: The intended texture of the media, such as Antique, Calenared, Linen, Stipple or Vellum.
   - **Grade**: The intended grade of the media, such as Gloss-coated paper, Uncoated white paper, etc.
   - **Hole Name**: A predefined hole pattern that specifies the pre-punched holes in the media, such as R2-generic, R2m-MIB, R4i-US, etc.
3. Click **OK**.

Rename Media

To rename Media:

- On the **Resources** pane, expand the **Contexts** folder, expand the **Media** folder, right-click the Media and click **Rename**. Type the new name and click **OK**.
- Alternatively, on the **Resources** pane, expand the **Contexts** folder, expand the **Media** folder, right-click the Media and click **Properties**. Type the new name in the **Name** field and click **OK**.

Applying Media to a page in a Print section

Every page in a print section has a natural position: it can be the first page, the last page, one of the pages in between (a 'middle page'), or a single page. For each of those positions, you can set different Media.

To apply Media to specific page positions in a Print section:

1. On the **Resources** pane, expand the **Print** context; right-click the Print section, and click **Sheet configuration**.
2. Optionally, check **Duplex** to enable content to be printed on the back of each sheet. Your printer must support duplex for this option to work. If Duplex is enabled, you can also check **Tumble** to duplex pages as in a calendar, and **Facing pages** to have the margins of the section switch alternately, so that pages are printed as if in a magazine or book.
3. If the option **Same for all positions** is checked, the same Media will be applied to every page in the print section. Uncheck this option.
4. Decide which Media should be linked to each sheet position: click the downward pointing arrow after **Media** and select a Media.
5. Optionally, decide which Master Page should be linked to each sheet; see "Master Pages" on page 156.

Note

When both Media and a Master Page are used on a certain page, they will both be displayed on the Preview tab of the workspace, the Master Page being 'in front' of the Media and the Print section on top. To open the Preview tab, click it at the bottom of the Workspace or select **View > Preview View** on the menu.

Note

Master Pages, Media and Duplex printing options can also be set in a Control Script (see "Control Scripts" on page 429 and "Control Script API" on page 931). This is especially useful when you need identical sections with different settings.

Dynamically switching the Media

In addition to applying Media to sheets via the settings, it is possible to change Media dynamically, based on a value in a data field, in a script. The script has already been made; you only have to change the name of the Media and the section in the script, and write the condition on which the Media has to be replaced.

1. On the **Resources** pane, expand the **Contexts** folder, expand the **Print** context, right-click the print section and click **Sheet configuration**.

2. Decide which pages should have dynamically switching media: every first page in the Print section, every last page, one of the pages in between (a 'middle page'), or a single page. (Uncheck the option **Same for all positions**, to see all page positions.)
3. In the area for the respective sheet position, click the **Edit script** button next to **Media**. The Script Wizard appears with a standard script:
   ```
   results.attr("content","Media 1");
   ```
   Media 1 will have been replaced with the name of the media selected for the chosen sheet position.
   The field **Selector** in the Script Wizard contains the name of the section and the sheet position that you have chosen.
4. Change the script so that on a certain condition, another media will be selected for the content. For instance:
   ```
   if(record.fields.GENDER === 'M') {
               results.attr("content","Media 2");
               }
   ```
   This script changes the media to Media 2 for male customers.
   See "Writing your own scripts" on page 393 if you are not familiar with how scripts are written.
5. Click **Apply**, open the tab **Preview** and browse through the records to see if the script functions as expected.
6. When you click **OK**, the script will be added to the **Scripts** pane.

Rotating the Media in a Print section

The actual orientation of the Media and that of a section to which the Media is applied may not match.
The Media (to be more accurate: the Virtual Stationery images specified for this Media) can therefore be rotated per Print section:

- On the **Resources** pane, expand the **Print** context; right-click the Print section, and click **Sheet configuration**.
- Click one of the options next to **Media rotation**.

The Media will be rotated accordingly in the entire section.

Note

- Any Virtual Stationery settings made for the Media also influence how the Media is displayed in each section (see "Setting Media properties" on page 160).
- Section backgrounds are rotated separately (see "Using a PDF file or other image as background" on page 142).

If in the Media properties, the Virtual Stationery position is set to Absolute, any offset given by the Top and Left values will be applied **after** rotation. A Virtual Stationery image located absolutely at the top left (Top: 0, Left: 0) will still appear at the top left of the page after rotating the Media.

Printing virtual stationery

Media are not printed, unless you want them to. Printing the virtual stationery is one of the settings in a Job Creation Preset. To have the virtual stationery printed as part of the Print output:

1. Create a job creation preset that indicates that Media has to be printed: select **File > Presets** and see "Output Creation Presets Wizard" on page 700 for more details.
2. Select that job creation preset in the Print Wizard; see "Generating Print output" on page 993.

Email

With the Designer you can create one or more Email templates and merge the template with a data set to generate personalized emails.

The Email **context** is the folder in the Designer that can contain one or more Email templates, also called Email **sections**. The HTML generated by this context is meant to be compatible with as many clients and as many devices as possible.

Email template

It is strongly recommended to start creating an Email template with a Wizard; see "Creating an Email template with a Wizard" on page 171.
Also see "Designing an Email template" on the next page for guidelines on the design.

Designing HTML email that displays properly on a variety of devices and screen sizes is challenging. Building an email is not like building for the web. While web browsers comply with standards (to a significant extent), email clients do not. Different email clients interpret the same HTML and CSS styles in totally different ways.

When an Email template is created, either with a Wizard or by adding an Email context to an existing template (see "Adding a context" on page 119), the Email context folder is created along with other files that are specific to an Email context; see "Email context" on page 174.

Only one Email section is created at the start, but you can add as many Email sections as you need; see "Email templates" on page 177. However, when the Designer merges a data set to generate output from the Email context, it can merge only one of the templates with each record; see "Generating Email output" on page 1017.

Email templates are personalized just like any other template; see "Variable Data" on page 351.

Sending email

When the template is ready, you can change the email settings (see "Email header settings" on page 180) and send the email. To test a template, you can send a test email first.

Output, generated from an Email template, can have the following attachments:

- The contents of the Print context, in the form of a single PDF attachment. (Compression options for PDF attachments can be specified in the Email context's properties; see "Compressing PDF attachments" on page 176.)
- Other files, an image or a PDF leaflet for example.

Attaching the Print context is one of the options in the "Send (Test) Email" on page 543 dialog.

See "Email attachments" on page 187 and "Generating Email output" on page 1017.

Designing an Email template

With the Designer you can design Email templates. It is strongly recommended to start creating an Email template with an Email Template Wizard, because it is challenging to design HTML email that looks good on all email clients, devices and screen sizes that customers use when they are reading their email.

This topic explains why designing HTML email design is as challenging as it is, which solutions are used in the Email Template Wizards and it lists good practices, for example regarding the use of images in HTML email. It will help you to create the best possible Email templates in the Designer.

HTML email challenges

Creating HTML email isn't like designing for the Web. That's because email clients aren't like web browsers. Email clients pass HTML email through a preprocessor to remove anything that could be dangerous, introduce privacy concerns or cause the email client to behave unexpectedly. This includes removing javascript, object and embed tags, and unrecognized tags. Most preprocessors are overly restrictive and remove anything with the slightest potential to affect the layout of their email client. Next, the HTML has to be rendered so that it is safe to show within the email client. Unfortunately, desktop, webmail, and mobile clients all use different rendering engines, which support different subsets of HTML and CSS. More often than not, the result of these operations is that they completely break the HTML email's layout.

Designing HTML email in PrintShop Mail Designer

The problem of HTML email is that preprocessing and rendering engines break the HTML email's layout. HTML tables, however, are mostly left untroubled. As they are supported by every major email client, they are pretty much the only way to design HTML emails that are universally supported. That's why Tables are heavily used to position text and images in HTML email.

Nesting tables (putting tables in table cells) and applying CSS styles to each table cell to make the email look good on all screen sizes is a precision work that can be a tedious and demanding. Connect's Designer offers the following tools to make designing HTML email easier.

Email templates: Slate and others

The most obvious solution offered in the Designer is to use one of the templates provided with the Designer; see "Creating an Email template with a Wizard" on page 171. The layout of these templates has been tested and proven to look good in any email client, on any device and screen size. The Tables in these templates are nested (put inside another table) and they have no visible borders, so readers won't notice them.

> **Tip**
>
> Click the **Edges** button on the toolbar to make borders of elements visible on the Design tab. The borders will not be visible on the Preview tab or in the output.

Litmus

There are several tools to preview how email will be rendered on a variety of clients. We recommend using **Litmus**. Support for Litmus is integrated into the Designer; the Send Test Email dialog has an option to "Send to Litmus".

Emmet

Emmet is a plugin that enables the lightning-fast creation of HTML code though the use of a simple and effective shortcut language. The Emmet functionality is available in the HTML and CSS source editors of Connect Designer. Emmet transforms abbreviations for HTML elements and CSS properties to the respective source code. The expansion of abbreviations is invoked with the **Tab** key.

In the Source tab of the Workspace, you could for example type `div.row`. This is the abbreviation for a element with the class row.
On pressing the Tab key, this abbreviation is transformed to:
`<div class="row"></div>`
To quickly enter a table with the ID 'green', one row, and two cells in that row, type:
`table#green>tr>td*2`
On pressing the Tab key, this is transformed to:

```
<table id="green">
        <tr>
                <td></td>
                <td></td>
        </tr>
</table>
```

All standard abbreviations can be found in Emmet's documentation: Abbreviations.

To learn more about Emmet, please see their website: Emmet.io and the Emmet.io documentation: http://docs.emmet.io/.

Preferences

To change the way Emmet works in the Designer, select **Window > Preferences**, and in the Preferences dialog, select **Emmet**; see "Emmet preferences" on page 382.

Using CSS files with HTML email

Email clients do not read CSS files and some even remove a <style> tag when it is present in the email's header. Nevertheless, CSS files can be used with the Email context in the Designer.
When generating output from the Email context, all CSS rules that apply to the content of the email can be processed and added either to the header of the email or to inline style properties as if local formatting was applied, depending on the Email section properties. See "Email section properties" on page 540.

Using images in email campaigns: tips

Host images on a public server

In the Designer you can add images as resource to the template document. When used in email messages these images are automatically embedded on sending the email. These embedded images appear instantly when viewing the message in your email client.
There is, however, a downside to this method: embedded images can't be used to track email open rates. Email services like mandrillapp.com embed a tiny tracer image at the bottom of your message. Each time a recipient opens the email the tracer image (aka beacon image) is downloaded and yet another 'open' is registered. On mobile devices this happens when the user clicks the Display Images button.
So, when tracking open rates in your email campaigns, store your images on a publicly-accessible server (preferably your own server) or a reputable image hosting service, like photobucket.com. Don't forget to set the Alternate Text for your images on the Attributes pane.

Do not capture your email in one big image

Most e-mail clients do not automatically download images, so do not capture your email in one big image. The recipient initially sees a blank message and probably deletes it right away.

Do not resize images in your email

Many mail clients do not support image resizing and will show the image in its original dimensions. Resize the images before you link to or embed them.

Use background images wisely

Most mail clients do not support background images: a very good reason to stay away from them in your mainstream email campaign. There is one situation in which they do come in handy. Both iPhone and Android default mail have solid CSS support and cover most of the mobile marketspace. You could use background images to substitute images when viewed on these devices. This is done by hiding the actual image and showing a mobile-friendly image as background image instead. This is a technique used in Responsive Email Design.

Creating an Email template with a Wizard

With the Designer you can design Email templates as well as PDF attachments. PDF attachments are designed in the Print context; see "Print context" on page 132.

It is strongly recommended to start creating an Email template with a Wizard, because designing HTML email that displays properly on a variety of devices and screen sizes is challenging. Email clients can, and will, interpret the same HTML and (inline) CSS in totally different ways (see "Designing an Email template" on page 167).

With an Email Template Wizard you can easily create an Email template that outputs emails that look good on virtually any email client, device and screen size.

After creating an Email template, the other contexts can be added to it, as well as other sections (see "Contexts" on page 118 and "Email templates" on page 177).

To create an Email template with a Template Wizard:

1. In the **Welcome** screen that appears after startup:
    - Choose **Browse Template Wizards**.
      Scroll down until you see the Email Template Wizards. There are three types of Email Template Wizards:
        - Basic Email templates
        - Banded Email templates
        - Slate: Responsive Email templates by Litmus.
    - Or choose **Create a New Template** and select the Email template. This starts the Basic Action Email wizard.

    Alternatively, on the **File** menu, click **New**, and:

- Select Email Template. This starts the Basic Action Email wizard.
- Or expand the **Template** folder, and then expand the **Basic Email templates** folder, the **Banded Email templates** folder, or the **Slate: Responsive Email Templates by Litmus** folder.

See "Email Template Wizards" on the next page for information about the various types of Template Wizards.

2. Select a template and click **Next**. If you don't know what template to choose, see below; the characteristics of each kind of template are described further down in this topic.
3. Make adjustments to the initial (or last used) settings. The options for each type of template are listed below.
   Click **Next** to go to the next settings page if there is one.
4. Click **Finish** to create the template.

The Wizard creates:

- An Email context with one section in it. The section contains dummy text and one or more **selectors** for variable data, for example: "Hello @first@". You will want to replace those by the names of fields in your data. See "Variable Data" on page 351.
- One **script**, named "To". Double-click that script on the **Scripts** pane to open it. This script ensures that the email is sent to an email address that is specified in a data field called "email-to". After loading data, you can change the script so that it uses the actual field in your data that holds the customer's email address. See "Email header settings" on page 180
- A style sheet, named context_htmlemail_styles.css, and another style sheet depending on which Template Wizard was used. The style sheets can be found in the **Stylesheets** folder on the **Resources** pane.

The Wizard opens the Email section, so that you can fill it with text and other elements; see "Content elements" on page 191, "Email context" on page 174, and "Email templates" on page 177.

Tip

Use the **Outline** pane at the left to see which elements are present in the template and to select an element.

Use the **Attributes** pane at the right to see the current element's ID, class and some other properties.

Use the **Styles** pane next to the Attributes pane to see which styles are applied to the currently selected element.

Note that the contents of the email are arranged in tables. The many tables in an Email template ensure that the email looks good on virtually any email client, device and screen size. As the tables have no borders, they are initially invisible.

Tip

Click the **Edges** button on the toolbar t emporarily adds a frame to certain elements on the Design tab. These will not .Print or output.

Email Template Wizards

There are Wizards for three kinds of Email templates: for **Basic Email**, for **Banded Email**, and **Slate** templates for responsive email designed by Litmus.

Slate: Responsive Email Templates by Litmus

Scroll past the Web Template Wizards to see the Slate: Responsive Email templates, created by Litmus (see https://litmus.com/resources/free-responsive-email-templates).

More than 50% of emails are opened on mobile. These five responsive HTML email templates are optimized for small screens and they look great in any inbox. They've been tested in Litmus and are completely bulletproof.

Tip

After creating the email template, click the Responsive Design View icon at the top of the workspace to see how the email looks on different screen sizes.

The only thing you can set in advance for a Slate template is the color of the call-to-action button. Click the small colored square, right next to the field that holds the default color value, to open the Color dialog and pick a color (see "Color Picker" on page 492).
The color can be changed later; see "Colors" on page 328.

Basic Email and Banded Email

The difference between Basic and Banded email is that the contents of a Basic email extend to the email's margin, rather than to the edge of the window in which it is read, as the contents of Banded emails do.
The Banded Email **Action** Template is a simple call-to-action email with a message, header and a button linking to a website, such as an informational or landing page.
The Banded Email **Invoice** Template is an invoice with an optional Welcome message and Pay Now button.

Settings

For a **Blank** email you can not specify any settings in the Wizard.

For an **Action** email, the Email Template Wizard lets you choose:

- The subject. You can change and personalize the subject later, see "Email header settings" on page 180.
- The text for the header. The header is the colored part at the top. The text can be edited later.
- The color of the header and the color of the button. Click the small colored square, right next to the field that holds the default color value, to open the Color dialog and pick a color (see "Color Picker" on page 492). The color can be changed later; see "Colors" on page 328.
- The web address where the recipient of the email will be taken after clicking the button in the email. Type the URL in the **Link** field.

Email context

In the Designer the Email context is the folder that contains Email templates. From the Email context, output can be generated in the form of email (see below).

When an Email template is created (see "Creating an Email template with a Wizard" on page 171) or when an Email context is added to a template (see "Adding a context" on page 119) the following happens:

- The Email context is created and one **Email section** is added to it. You can see this on the **Resources** pane: expand the **Contexts** folder, and then expand the **Email** folder. See "Email templates" on page 177 to learn how to fill an Email section.
  Although only one email can be sent per record when generating Email output, the Email context can contain multiple sections. One Email section is created at the start, but you can add more; see "Adding an Email template" on page 177 and "Importing an Email template" on page 178.
- A style sheet, named context_htmlemail_styles.css, is added to the template. Depending on which Template Wizard was used to create the template, another style sheet can be added as well. Style sheets are located in the folder **Stylesheets** on the **Resources** pane. These style sheets are meant to be used for styles that are only applied to elements in the Email context.

The Wizard opens the Email section, so that you can fill it with text and other elements; see "Content elements" on page 191 and "Email templates" on page 177.

Sending email

When the template is ready, you can generate Email output; See "Generating Email output" on page 1017.
To test a template, you can send a test email first. This allows you to override the recipient address.

Output, generated from an Email template, can have the following attachments:

- The contents of the Print context, in the form of a single PDF attachment. (Compression options for PDF attachments can be specified in the Email context's properties; see "Compressing PDF attachments" on the facing page.)
- Other files, an image or a PDF leaflet for example.

Attaching the Print context is one of the options in the "Send (Test) Email" on page 543 dialog.

Note

To split the Print context into multiple attachments, and to rename this type of attachment, you need to create a Control Script that specifies **parts**; see "Parts: splitting and renaming email attachments" on page 436.

See "Email attachments" on page 187.

Email output settings

The following settings in an Email context influence how the Email output is generated.

Compressing PDF attachments

For PDF attachments, generated from the Print context, you can set the Print Context Image Compression to determine the quality of the files, and with that, the size of the files.

To set the Print Context Image Compression:

1. On the **Resources** pane, expand the **Contexts** folder; then right-click the **Email** context and select **PDF Attachments**.
   Alternatively, select **Context > PDF Attachments** on the main menu. This option is only available when editing an Email section in the Workspace.
2. Change the properties of the PDF file that will be attached when the Print context is attached to the email.
   Lossless is the maximum quality. Note that this will produce a larger PDF file. Uncheck this option to be able to set a lower quality.
   The **quality** is set in a percentage of the maximum quality.
   Tile Size is the size of the files in which the image that is being compressed is divided. (If the image height or width is not an even multiple of the tile size, partial tiles are used on the edges.) Image data for each tile is individually compressed and can be individually decompressed. When low Quality values are used to optimize images smaller than 1024 x 1024 pixels, using the largest tile size will produce better results.

Setting a default section for output

When generating output from the Email context, only one of the Email templates can be merged with each record. One of the Email sections is the 'default'; see "Setting a default Email template for output" on page 180.

Email templates

Email templates (also called Email **sections**) are part of the Email context in a template. The Email context outputs HTML email with embedded formatting to an email client through the use of an email server. Since email clients are numerous and do not support same features, the HTML generated by this context is not optimized for any specific client - rather, it's meant to be compatible with as many clients and as many devices as possible.

In Email templates, many content elements can be used; see "Content elements" on page 191. However, special attention must be paid to the way elements are positioned. In Email sections, it is advisable to position elements using Tables and to put text in table cells (see "Designing an Email template" on page 167).

Email templates are personalized just like any other template; see "Variable Data" on page 351.

The subject, recipients (To, CC and BCC), sender and reply-to address are specified with Email Script Wizards; see "Email header settings" on page 180.

An Email context can contain multiple templates. When generating output from the Email context, however, only one of the Email templates can be merged with each record. Set the 'default' Email section (see below) before generating Email output; see also "Generating Email output" on page 1017.

For information about attachments see "Email attachments" on page 187.
A plain-text version of the HTML is added to each email if the option is checked in the Email section's properties (see "Properties tab" on page 541). With new templates this is always the case.

Adding an Email template

When an Email template is created (see "Creating an Email template with a Wizard" on page 171), only one Email section is added to it. An Email context may contain various templates, but per record only one of those can be sent when you generate Email output.

It is not possible to add an Email section to an existing Email context with the help of a Template Wizard.

To provide alternative content for your email, you could use Conditional Content (see "Showing content conditionally" on page 358), or Snippets and a script (see "Snippets" on page 285 and "Loading a snippet via a script" on page 420).

If you would like to start with a template that is identical to the one you already have, consider copying it (see "Copying a section" on page 120). If it's inside another template you can import it (see below).

To add a section to the Email context:

- On the **Resources** pane, expand the **Contexts** folder, right-click the **Email** folder, and then click **New Email**.

Importing an Email template

To import an Email section from another template, click **File > Import Resources...** in the menu. See: "Import Resources dialog" on page 502.
Remember also to add or import any related source files, such as images.

Note that when the imported Email section *replaces* an Email section in your template, the PDF attachments settings are imported as well. (See: "Compressing PDF attachments" on page 176.)

Deleting an Email template

To delete an Email section:

- On the **Resources** pane, expand the **Contexts** folder, expand the **Email** context, right-click the name of the section, and then click **Delete**.

> **Warning**
>
> If you don't have a backup of the template, the only way to recover a deleted section, is to click **Undo** on the **Edit** menu, until the deleted section is restored. After closing and reopening the template it is no longer possible to restore the deleted context this way.
> In the Saving Preferences you can set whether a backup file should be created when you save the template; see "Save preferences" on page 391.

Styling and formatting an Email template

The contents of an Email section can be formatted directly, or styled with Cascading Style Sheets (CSS). See "Styling and formatting" on page 294.

Email clients do not read CSS files and some even remove a <style> tag when it is present in the email's header. Nevertheless, CSS files can be used with the Email context in the Designer.
When generating output from the Email context, all CSS rules that apply to the content of the email can be processed and added either to the header of the email or to inline style properties as if local formatting was applied, depending on the Email section properties. See "Email section properties" on page 540.

> **Tip**
>
> Before you can style an element, you have to select it. In an Email context it can be difficult to select an element by clicking on it. Use the **breadcrumbs** at the top and the **Outline** pane at the left, to select an element. See "Selecting an element" on page 196.

In order for a style sheet to be applied to a specific section, it needs to be included in that section. There are two ways to do this.

Drag & drop a style sheet

1. Click and hold the mouse button on the style sheet on the **Resources** pane.
2. Move the mouse cursor within the **Resources** pane to the section to which the style sheet should be applied.
3. Release the mouse button.

Using the Includes dialog

1. On the Resources pane, right-click the section, then click **Includes**.
2. From the **File types** dropdown, select **Stylesheets**.
3. Choose which CSS files should be applied to this section. The available files are listed at the left. Use the arrow buttons to move the files that should be included to the list at the right.

4. You can also change the order in which the CSS files are read: click one of the included CSS files and use the **Up** and **Down** buttons. Note that moving a style sheet up in the list gives it **less** weight. In case of conflicting rules, style sheets read later will override previous ones.

> **Note**
>
> Style sheets are applied in the order in which they are included in a section. The styles in each following style sheet add up to the styles found in previously read style sheets. When style sheets have a conflicting rule for the same element, class or ID, the **last** style sheet 'wins' and overrides the rule found in the previous style sheet.

> **Note**
>
> Style sheets that are linked to (i.e. included in) a section show a chain icon in the Resources pane (see "Resources pane" on page 583).

Setting a default Email template for output

An Email context can contain multiple templates. When generating output from the Email context, however, only one of the Email templates can be merged with each record.

To select the Email section that will be output by default:

- On the **Resources** pane, expand the **Email** context, right-click a section and click **Set as Default**.

> **Tip**
>
> Use a Control Script to dynamically select an Email section for output depending on the value of a data field.

Email header settings

An email header contains routing information, such as the **sender**, **recipient**/s and **subject** of the message. This topic explains how to customize the header of an email that is generated

from an Email template.

The default Email SMTP settings and the sender's name and address are defined in the Connect Designer preferences (see "Email preferences" on page 380). They can be adjusted per run in the Send Email and Send Test Email dialogs.

Email Fields

The **subject**, the **recipients** (To, Cc and Bcc), the **sender** and the **Reply to** address can be entered in the **Email Fields** at the top of the workspace. If the fields are not visible, click the words 'Email Fields' (or the small plus before them) to expand the Email Fields area.
To use a variable email address in any of the fields, simply **drag and drop** a data field into the email field.
The specified subject and addresses will be visible when viewing the email in the workspace in Preview mode.
The To address must always be variable. This field is not used when you send a test email (see "Generating Email output" on page 1017).

> **Note**
>
> Using a variable email address requires you to load data first; see "Loading data" on page 341.

The Email Script Wizard

In addition to the drag and drop method, you can use the Email Script Wizard to add data to an email header field. It lets you choose one or more data fields and enter a prefix and/or suffix (per data field).
There are two ways to open the Email Script Wizard:

- Via the **Email Fields**. Open the email section and expand the Email Fields at the top by clicking **Email Fields**. Click the word before the email field that you want to set. If there already is a script for that field, that script will be opened. Otherwise, a new script will be created and opened.
- Via the **Scripts** pane. Click the black triangle on the **New** button and select the respective email script. A new script will be added to the Scripts pane. Double-click the new script to open it. See "Script wizards" on page 534 for an explanation of the options in the script wizard.

The default script adds the content of the selected data field to the header field.
If you want to write a more complex script, click the **Expand** button. The result of the script should be a valid, fully-formed email address.
The language in which the script has to be written is JavaScript. For more information on writing scripts, see "Writing your own scripts" on page 393.

Other header fields

At some point you may need to define a header field that isn't available in the Preferences or in the Email Fields. This can be done in a Control Script. For a few examples of such scripts, see "Adding custom ESP handling instructions" on page 1023. To get started with Control Scripts, refer to "Control Scripts" on page 429.

Email SMTP settings

Simple Mail Transfer Protocol (SMTP) is the standard protocol for sending emails across the Internet.

Default SMTP settings can be specified in the Preferences dialog: select **Window > Preferences**, expand the **Email** preferences and click **SMTP**.

You can add as many presets as needed, for example for different Email Service Providers (see "Using an ESP with PrintShop Mail Connect" on page 1021). To do this, click the Add button at the right. Then fill out the following settings:

- **Name**: The name of the preset. This will show up in the Send Email dialog.
- **Host**: The SMTP server through which the emails are to be sent. This can be a host (mail.domain.com) or an IP address.

> **Tip**
>
> Gmail only allows Connect to be used as an SMTP client when "Access for less secure apps" is enabled in the Google account settings.

- **Port**: You can specify a port number. This will be added to the host name, for example: smtp.mandrillapp.com:465.

> **Note**
>
> If the mail server supports it, the connection will be encrypted without the need to send the server a STARTTLS instruction when port 465 is used.

- **Use authentication**: Check this option and fill in the user name if a user name and password are needed to send emails through the host. (The password has to be specified in the Send Email or Send Test Email dialog.)
- **Send STARTTLS**: This option is enabled if authentication is checked. With STARTTLS the client negotiates with the mail server to use some form of encryption, usually a version of Transport Layer Security (TLS). Since this improves security it is recommend to enable this option if you use port 25 (the default port), 2525, or 587.
  Note that the email will not be sent if the SMTP server does not support TLS or SSL (an older encryption type).
  This option is ignored when port 465 is used.

When you click the **Restore** button, the presets for a number of Email Service Providers will appear.

> **Note**
>
> When updating the software from a version prior to version 1.5, pre-existing presets will be maintained in the new version.

In the Send Test Email dialog and Send Email dialog (see "Send (Test) Email" on page 543) you will be able to choose one of the presets and adjust the settings to your needs.

Subject

To specify a subject for an email template:

1. Open the email section and expand the Email Fields by clicking **Email Fields** at the top of the section.
2. Type the subject in the **Subject** field.

To add **variable data** to the subject of an email section, **drag and drop** a data field into the Subject field at the top of the workspace. Two things will happen:

- A placeholder for the data field appears in the subject line (for example: @email@).
- A new script, named Subject, is added to the Scripts pane.

You can add as many data fields to the subject as you like. When you do add more than one data field, the existing Subject script will be modified to include all data fields that are added to the subject.

The result of the script will be visible in the Subject field in Preview mode: click the Preview tab at the bottom of the workspace.

> **Note**
>
> By default, the Subject script targets one email section specifically. You can see this when you double-click the script on the Scripts pane. The selector of the Subject script contains the name of a particular email section, for example: html.HTML_EMAIL [section="Content"] (in this case, Content is the name of the email section). If you remove the [section=...] part from the selector, the script will work for all email sections.
>
> Subject scripts made with versions of the software prior to version 1.7 are **not** specific to one email section.

Writing a custom Subject script

The default script replaces all @field@ placeholders in the subject line with field values. This script can be modified, for example to create a subject that depends on the value of a data field. Open the Script Wizard (see "The Email Script Wizard" on page 181), click the **Expand** button and modify the script.
If you don't know how to write a script, see "Writing your own scripts" on page 393 first.

> **Note**
>
> A Subject script created by clicking Subject in the Email Fields always targets one email section specifically, for example: html[section="Content"] (in this case, Content is the name of the email section). Remove the `[section=...]` part from the **selector** of the script to make the script work for all email sections.

Recipients: To, CC and BCC

To specify recipients for Email output, you can simply **drag and drop** a data field that contains an email address into the **To** field at the top of the workspace. A new script, named To, will be added to the Scripts pane.
Note that you can add only one data field to the email field this way. When you drag another data field into the email field the existing script will be replaced..

Email addresses can be added to the **Cc** and **Bcc** fields in the same manner, but it is also possible to type an email address directly in the Cc or Bcc field (as long as no script is present for that field). Multiple email addresses should be separated by a semicolon.
Email addresses in the Bcc ('blind carbon copy') field will not be visible to any other recipient of the email.

Alternatively, you could use the Script Wizard to create the scripts; see "The Email Script Wizard" on page 181.

Sender

From address

A default **From** name and email address can be specified in the Preferences dialog: select **Window > Preferences**, expand the **Email** preferences and click **General**.
This name and email address will appear as the default in the Send Email and Send Test Email dialogs (see "Send (Test) Email" on page 543).

The default can be overwritten by typing an email address directly in the From field (as long as no script is present for this field).

Using the Script Wizard you can create a **dynamic** From address; see "The Email Script Wizard" on page 181. It is also possible to drag and drop one data field into the From field directly.

> **Tip**
>
> A dynamic From address is often used when sending email campaigns and to do tracking of email replies. Include the recipient's email address in a dynamic From address to enable automatic detection and removal of undeliverable e-mail addresses. (This technique is called VERP; see Wikipedia.)

Reply To address

The Reply To address is used by mail clients, when the recipient clicks the Reply To (or Reply All) button.

You can type an email address directly in the Reply To field (as long as no script is present for this field).
Alternatively, you can drag and drop one data field into the field, or use the Email Script Wizard (see "The Email Script Wizard" on page 181), to specify the Reply To address in a script.

Meta information

Meta information in an email header is not visible to the receiver, but might have an effect on the representation of the email in the e-mail client. The information may also be used by an Email Service Provider (see "Using an ESP with PrintShop Mail Connect" on page 1021).

To add custom meta information to an email header:

1. Right-click the email section on the Resources pane and select **Properties...**.
2. The **Meta Information** group lists all <meta> tags that will be added to the header of the email. Click the **Add** button to add a new <meta> tag to the list.
3. Select the **type** of <meta> tag, which is either name or http-equiv.
4. For a name-type meta tag, enter the **value**.
5. Enter the **content**.

Example

When you add a *name* meta tag with the value *viewport* and content *width=320*, the following will be added to the email header:
<meta name="viewport" content="width=320">.

For more information on <meta> tags, see W3Schools - HTML meta tag.

Email PDF password

The Email PDF Password Script Wizard defines a password with which to protect the PDF generated when using the Print context as PDF Attachment option in the Send Email or Send Test Email dialogs (see "Generating Email output" on page 1017). The result of the script will be the password necessary to open the PDF when it is received by email.

To define a password to protect the generated PDF attachment:

1. On the Scripts pane, click the black triangle on the **New** button and click **Email PDF password Script**. A new script is added to the Scripts pane.
2. Double-click the new script to open it.
3. Select a data field and optionally, type a prefix and/or suffix.

Password types

PDF allows for two types of passwords to be set on a secured PDF file: a user password and owner password. The user password allows a limited access to the file (e.g. printing or copying text from the PDF is not allowed). The owner password allows normal access to the file. The Email PDF password script sets both the user and owner password to the same value, so that when the recipient provides the password, he can manipulate the file without limitations.

> **Note**
>
> If a template has a Control Script that creates multiple PDF attachments, all the attachments are secured by the same password.

> **Note**
>
> Via a Control Script it is possible to set a different user password and owner password, see "Control Script: Securing PDF attachments" on page 443, "Control Scripts" on page 429 and "Control Script API" on page 931.

Email attachments

Output, generated from an Email template, can have the following attachments:

- The contents of the Print context, in the form of a single PDF attachment. (Compression options for PDF attachments can be specified in the Email context's properties; see "Compressing PDF attachments" on page 176.)
- Other files, an image or a PDF leaflet for example.

Attaching the Print context is one of the options in the "Send (Test) Email" on page 543 dialog.

Attaching a Context

By default, when adding the Print context to an email, all Print sections are output to a single PDF file, named after the email subject, which is then attached to the email.
The PDF can be protected with a password (see "Email PDF password" on page 186).
Compression options for PDF attachments can be specified in the Email Context Properties dialog; see "Compressing PDF attachments" on page 176.

> **Note**
>
> To split the Print context into multiple attachments, and to rename this type of attachment, you need to create a Control Script that specifies **parts**; see "Parts: splitting and renaming email attachments" on page 436.

Attaching files

Selecting and adding files as attachments

If you want all recipients to get the same attachments with their email, you can add the attachments to the Email section(s).

The easiest way is to **drag and drop** the desired file on the Email section.
If the file is an image, you will be presented with the option to import it into the template's Resources folder. Any other file will be added to the list of attachments directly.

The **Attachments** dialog also lets you select files and delete attachments.
To open the Attachments dialog, **right-click** the Email section in the Resources pane and select **Attachments**.
Alternatively, select **Section > Attachments** from the menu. This menu item is only available when an Email section is opened in the workspace.

For further information about this dialog, see: "Attachments dialog" on page 477.

Dynamic attachments: creating file names based on data fields

The **Dynamic Attachment Script Wizard** lets you add a different attachment for each email recipient. It composes **one** file name (including the path) based on the value of one or more data fields.

The Dynamic Attachment wizard lets you concatenate this value with the base location and/or file extension to construct the path. Dynamic Attachment scripts are created via the New option on the toolbar of the Scripts panel.

1. On the **Scripts** pane at the bottom left, click the downward pointing arrow next to the **New** button; then select **Email Scripts > Dynamic Attachment**.

   > **Note**
   >
   > Note that an Attachments script creates **one single attachment**. To add more attachments, you could either add Attachments scripts, or click the Expand button and edit the script.

2. A new script called **Attachments** has appeared in the list. Double-click to open it.
3. Choose an Email section from the **Section** drop-down list.
4. Fill in the different parts of which the file name is composed:
   - **Prefix**. The first prefix contains the base path (or at least the first, static part of the path). For example:`C:\Attachments\`, `C:/Attachments/`, or `file:///C:/Attachments/`.
   - **Data field/s**. The selected data field/s will be evaluated. If a data field is empty, the entire row is skipped. Otherwise the prefix, data field value and suffix are added to the path/file name.
   - **Suffix**. The suffix on the last used row should contain the file extension, including the dot (for example `.pdf`).

   For resources **inside** the template, refer to the folder in the Resources, e.g. 'images/file.extension' , or 'fonts/myfont.otf', etc.
   For any other file, give a valid **URL**.
   - Use the file protocol for a file on **disk**, for example: file:///c:/somefolder/attachments/INV2018.246.pdf (which equals file://localhost/c:/somefolder/attachments/INV2018.246.pdf; if the host is "localhost", it can be omitted).
   - For a **remote** file, you can use the http protocol e.g. http://www.mysite.com/somefolder/attachments/INV2019-246.pdf or http://localhost:8080/pod/v1/deliverynotes/{8FCEC8BC-72E8-486B-A206-516BF10E21F6}.

Note

For attachment names, it is recommended to use only US-ASCII characters. Other characters may not be supported by all email servers and clients.

Note

Certain characters are invalid in a URL (for example, '$', '%', and '&') and must be percent-encoded. The same applies to a file path, since that actually is a URL that starts with the file protocol.

Note that even a space character is invalid in a URL. Spaces in a URL are supported for backward compatibility, but it is recommended to percent-encode a space character as %20.

5. The attachment's name in the email will be the part of the path that comes after the last '/'. When there are no forward slashes in the path, the full path is used.
   You may want to use a custom attachment name. To learn how to do that, see "Renaming attachments" below.
6. Click **OK** or **Apply** to save your changes.

Note that an Attachments script creates **one single attachment**. To add more attachments, you could either add Attachments scripts, or click the Expand button and edit the script.

If you want to write your own email attachment scripts, there is a how-to that you may find helpful: How to add custom email attachments.

Renaming attachments

Sections that are attached to an email can only be renamed via a Control Script; see "Parts: splitting and renaming email attachments" on page 436.

Renaming dynamic attachments

Dynamic attachments can be renamed via the script that attaches them to the email. Double-click the script to open it and click the **Expand** button.

Dynamic attachment scripts add a <link> element to the <head> of an Email section. The title attribute of that element specifies the attachment name that will show up in the email.
Take a look at the last line of the script:
results.append(query("<link rel=related>").**attr("title", result.split('/').pop())**.attr("href", result));
To give the attachment another name, you have to replace the bold part of the code by that new name. For example:
results.append(query("<link rel=related>").attr("title", **"Invoice.pdf")**.attr("href", result));
Of course, you can also use data field values here, for example: results.append(query("<link rel=related>").attr("title", **record.fields.invoice_number + ".pdf")**.attr("href", result));

Note that the Wizard can no longer be used once you have edited and saved the script.

> **Note**
>
> For attachment names, it is recommended to use only US-ASCII characters. Other characters may not be supported by all email servers and clients.

Content elements

Once you have created a template, it can be filled with all kinds of elements. All types of elements are listed on this page.

There are several ways to insert elements, see "Inserting an element" on page 194.

Each element can have an ID and a class, as well as a number of other properties, depending on the element's type. When an element is selected, its properties can be changed; see "Selecting an element" on page 196, "Attributes" on page 193 and "Styling and formatting an element" on page 197.
ID's and classes are particularly useful with regard to variable data (see "Personalizing content" on page 338) and styling (see "Styling templates with CSS files" on page 296).

When you add elements, such as text, images or a table, to the content of a template, you are actually constructing an HTML file. It is possible to edit the source of the HTML file directly in the Designer; see "Editing HTML" on page 193.

Element types

The following types of content can be added to the content of a template:

- "Images" on page 271 and "Dynamic images" on page 366
- "Text and special characters" on page 283
- "Date" on page 267
- "Table" on page 278
- "Boxes" on page 256: Positioned Box, Inline Box, Div and Span

> **Tip**
>
> Wrapping elements in a box (see "Boxes" on page 256) or in a semantic HTML element makes it easier to target them in a script or in a style sheet. Place the cursor in the element or select multiple elements. Then, on the menu, click **Insert > Wrap in Box**. You can now use the wrapper element as a script's or style's selector; see "Using the Text Script Wizard" on page 353 and "Styling and formatting" on page 294.

- "Hyperlink and mailto link" on page 268
- "Barcode" on page 197
- "Whitespace elements: using optional space at the end of the last page" on page 149 (Print context only)
- "Page numbers " on page 150 (Print context only)
- Article, Section, Header, Footer, Nav and Aside are HTML5 semantic elements; see https://www.w3schools.com/html/html5_semantic_elements.asp
- Other HTML elements: Heading (H1 - H6), Address and Pre
  To quickly change a paragraph into a Heading, place the cursor inside of it, or select the paragraph (see: "Selecting an element" on page 196). Then select the appropriate element, either on the **Format** menu, or from the 'Element type' drop-down on the toolbar.
- "Snippets" on page 285: a Snippet is a small, ready-to-use piece of content in a file
- Business graphics

Most elements are suitable for use in all contexts. There are a few exceptions, however. Whitespace elements and Page numbers can only be used in a Print context. Positioned boxes are well suited for Print sections, but are to be avoided in the Email context.

Whether it is best to use a Table or Box to position text, images and other elements, depends on the context in which they are used; see "How to position elements" on page 312 for more information.

Editing HTML

When you add elements, such as text, images or a table, to the content of a template, you are actually constructing an HTML file.

To see this, toggle to the **Design** tab in the workspace. Click anywhere in the content. Take a look at the *breadcrumbs* at the top of the workspace or select the Outline pane. The breadcrumbs show the HTML tag of the clicked element, as well as the HTML tags of other elements to which the clicked element belongs. The clicked element is at the end of the line.

To edit the HTML text directly:

- In the workspace, toggle to the **Source** tab.

On this tab you can view and edit the content of the template in the form of plain text with HTML tags (note the angle brackets: <>). You may add and edit the text and the HTML tags, classes, ID's and other attributes.

To learn more about HTML, see for example https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Introduction and http://www.w3schools.com/html/default.asp.

Many video courses and hands-on courses about HTML (and CSS) are offered on the Internet as well, some for free. Go, for example, to www.codeschool.com or www.codeacademy.com and look for HTML (and CSS) courses.

Attributes

ID and class

Every element in the content of a template can have an **ID** and a **class**. ID's and classes are particularly useful with regard to variable data (see "Personalizing content" on page 338) and styling (see "Styling templates with CSS files" on page 296).

You can specify an ID and/or class when you add the element to the content.

To add an ID and/or class to an element that has already been added to a template, select the element (see "Selecting an element" on page 196) and type an ID and/or a class in the respective fields on the **Attributes** pane at the top right.

> **Note**
>
> Each ID should be unique. An ID can be used once in each section.

Other attributes

Apart from the ID and class, elements can have a varying number of properties, or 'attributes' as they're called in HTML (see "Editing HTML" on the previous page). Which properties an element has, depends on the element itself. An image, for example, has at least four attributes: `src` (the image's URL), `alt` (alternate text), `width` and `height`. These attributes are visible on the **Attributes** pane when you click an image in the content.

For each type of element, a small selection of attributes is visible on the **Attributes** pane at the top right.

In a multilingual template, the proprietary `data-translate` attribute marks an element for translation. For more information see "Translating templates" on page 467 and "Tagging elements for translation" on page 469.

Changing attributes via script

Many attributes can be changed via the user interface. Another way to change attributes is by using a script.

Any of the Script Wizards can produce a script that changes an attribute of an HTML element. Set the **Options** in the Script Wizard to **Attribute**, to output the script's results to the value of a specific attribute. See "Using the Text Script Wizard" on page 353.

In code, you can change an element's attribute using the function `attr()`; see "Writing your own scripts" on page 393 and "Standard Script API" on page 810.

Inserting an element

To insert an element in a section:

1. Navigate to where you want to insert the element, using the arrow keys, the mouse, the Breadcrumbs (see "Selecting an element" on the facing page) or the Outline pane.
2. Click the respective toolbar button. Alternatively, click the element on the **Insert** menu.
3. Add an ID and/or a class. ID's and classes are particularly useful with regard to variable data (see "Personalizing content" on page 338) and styling (see "Styling templates with CSS files" on page 296).

   Note

   Do not give an element the ID 'pages' or the class name 'dynamic'. These are reserved words. Using them as an ID or class name leads to undesirable effects.

4. Use the **Location** drop-down (if available) to select where to insert the element.
   - **At cursor position** inserts it where the cursor is located in the template.
   - **Before element** inserts it before the HTML element in which the cursor is currently located. For example if the cursor is within a paragraph, the insertion point will be before the tag.*
   - **After start tag** inserts it within the current HTML element, at the beginning, just after the start tag.*
   - **Before end tag** inserts it within the current HTML element, at the end, just before the end tag.*
   - **After element** inserts it after the element in which the cursor is currently located. For example if the cursor is within a paragraph, the insertion point will be after the end tag of the paragraph ( ).*
   - **Replace** inserts it in place of the currently selected element. (This option is not available when inserting content in a Master Page.)

   * If the current element is located inside another element, use the **Elements** drop-down to select which element is used for the insertion location. The list displays every element in the breadcrumbs, from the current selection point down to the root of the body.
   Note: HTML has some restrictions as to which types of elements are allowed as children of other elements. For example, a paragraph element is not allowed to have children that are block level elements - like a Div or a Table. If inserting content at the selected location would produce invalid HTML the final result may be different than expected. For example, when you insert a Div into a paragraph, the paragraph gets split in two. This means you end up with two paragraphs with the Div in between.

For a list of links to the different types of elements, see "Element types" on page 191.

Selecting an element

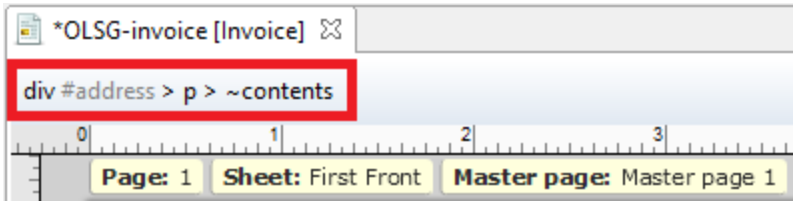
When an element is selected, the **Attributes** pane shows the attributes of that element, and the **Styles** pane, next to the **Attributes** pane, shows which styles are applied to it.

To select an element in the content, you can of course click on it, but this isn't always as easy as it seems, especially when the element has elements inside it.

> **Tip**
>
> Click the **Edges** button on the toolbar t emporarily adds a frame to certain elements on the Design tab. These will not .Print or output.

There are two more ways to select an element in the content:

- Using the *Breadcrumbs* at the top of the workspace.

  Breadcrumbs show the HTML tag of the clicked element, as well as the HTML tags of 'parent elements': elements inside of which the clicked element is located. The clicked element is at the end of the line.
  Elements with classes or IDs show these details next to them, for instance `div #contents > ol.salesitems > li ~contents`.
  Click any of the **elements** in the Breadcrumbs to select that element. If an element is selected in the Breadcrumbs and the Backspace key is pressed, that element is deleted.
  Click `~contents` to select the **contents** of the element. This way you may, for example, quickly change the text of a hyperlink.
- Using the **Outline** pane. You can find this pane next to the **Resources** pane. It displays a tree view of the elements in the file. Click an element in the tree view to select it.

Deleting an element

To delete an element, select it - as described above - and press the Delete key.
If the deleted element was targeted by a script, you will be asked if you want to delete the script as well.

Scripts are used to personalize templates. To start learning more about scripts, see "Personalizing content" on page 338 and "Writing your own scripts" on page 393.

Styling and formatting an element

Format elements directly

Images and other graphical elements can be resized by clicking on them and dragging the resize handles. There are toolbar buttons to color, indent or style text. Other toolbar buttons can left-align, right-align, or rotate graphical elements.

The toolbar buttons only represent a selection of the formatting options for each element. There are no toolbar buttons to change an element's margins, or to add a border to it, for example. To access all formatting properties of an element, you have to open the Formatting dialog. There are two ways to do this:

- Right-click the element and select the type of element on the shortcut menu.
- Select the element (see "Selecting an element" on the previous page) and select the type of element on the **Format** menu.

See "Styling and formatting" on page 294 for more information about the formatting options.

Format elements via Cascading Style Sheets (CSS)

It is highly recommended to use style sheets in templates right from the start. Even more so if the communications are going to be output to different output channels, or if they consist of different sections (for example, a covering letter followed by a policy). Using CSS with templates allows a consistent look and feel to be applied. A style sheet can change the look of multiple elements, making it unnecessary to format each and every element in the template, time and again, when the company's layout preferences change. See "Styling templates with CSS files" on page 296.

Barcode

In PrintShop Mail Connect Designer, you can add a variety of barcodes to your template. The supported Barcode types include 1d barcodes (the striped ones) and 2d barcodes (encoded horizontally and vertically).

Adding a Barcode

> **Note**
>
> When generating Print output, you can add extra barcodes and OMR marks. The reason why you would do this, is that at merge time more information is available about the actual output document. The page count, for example, is not available at design time. To add barcodes and OMR marks on the fly when generating Print output, select **File > Print** and check the option **Add additional content** (see "Additional Content" on page 732) in the Print Wizard. To have this done automatically, save this and other output options in an Output Creation Preset: select **File > Print Presets > Output Creation Settings** (see "Output Creation Presets Wizard" on page 700).

Before adding a Barcode, load data or at least a Data Model; see "Loading data" on page 341. You will need the field names when adding the Barcode. Then, to add a Barcode to a section, Master Page or snippet:

1. Select **Insert > Barcode** on the menu or click the **Barcode** toolbar button.
2. Choose the desired barcode type. The list is divided between 1d and 2d barcodes.
3. Give the barcode an **ID** and/or **class**.
4. Check the option **Absolute** to insert the barcode in an absolute-positioned box inside the <body> of the HTML, but outside other elements. (This option is not available in Email sections.) Alternatively, use the **Location** drop-down to select where to insert the Barcode.
    - **At cursor position** inserts it where the cursor is located in the template.
    - **Before element** inserts it before the HTML element in which the cursor is currently located. For example if the cursor is within a paragraph, the insertion point will be before the tag.*
    - **After start tag** inserts it within the current HTML element, at the beginning, just after the start tag.*
    - **Before end tag** inserts it within the current HTML element, at the end, just before the end tag.*
    - **After element** inserts it after the element in which the cursor is currently located. For example if the cursor is within a paragraph, the insertion point will be after the end tag of the paragraph ( ).*

- **Replace** inserts it in place of the currently selected element. (This option is not available when inserting content in a Master Page.)

* If the current element is located inside another element, use the **Elements** drop-down to select which element is used for the insertion location. The list displays every element in the breadcrumbs, from the current selection point down to the root of the body.
Note: HTML has some restrictions as to which types of elements are allowed as children of other elements. For example, a paragraph element is not allowed to have children that are block level elements - like a Div or a Table. If inserting content at the selected location would produce invalid HTML the final result may be different than expected. For example, when you insert a Div into a paragraph, the paragraph gets split in two. This means you end up with two paragraphs with the Div in between.

5. Under **Script**, select the field that contains the barcode value. The barcode type dictates the length and exact format of the required value. For a detailed description or for background information on a specific barcode, please refer to the documentation provided by the individual barcode supplier. Note that some barcode readers may require specific parameters as well.
If it is necessary to concatenate fields to compose the barcode value, edit the script after adding the barcode; see "Barcode script" on the facing page.

> **Note**
>
> For barcodes that require a Checksum, the Designer can calculate a Checksum if that isn't provided by your data. In that case the field should contain the required value minus the Checksum. To include a calculated Checksum in the barcode value, edit the barcode properties after adding the barcode to the template; see below.

6. Click **OK** to close the dialog.

In the template the barcode shows up as a gray box. The associated barcode script is added to the Scripts pane. To see the barcode script working, toggle to the **Preview** tab in the Workspace.

A barcode is always added with the barcode type's default properties and dimensions, but they can easily be changed; see "Barcode type and properties" on the facing page.

Changing a barcode

Barcode script

The barcode script determines which value is fed to the barcode generator. Double-click the script on the **Scripts** pane to change which field or fields are added to the barcode value. When you select more than one field, the script puts the values of the selected fields in one string and passes that to the barcode generator.

> **Tip**
>
> If you don't know which script matches the barcode, click the box that contains the barcode and check the ID and class of that box on the Attributes pane. Then take a look at the Scripts pane: the selector of the associated script is the same as the ID and/or class of the barcode box.
> If a barcode has more than one class name, only the first one is used in the script's selector.

Barcode type and properties

A barcode is always added with the barcode type's default properties and dimensions.

To change the barcode type or the barcode's properties such as the scale and color, open the Barcode properties dialog: right-click the barcode (on the **Design** tab in the Workspace) and select **Barcode** on the shortcut menu.

The barcode properties set via the properties dialog are written to the `data-params` attribute on the barcode element in JSON format. (To see this, select the barcode and open the document in the Source view.)

Click the barcode type below for information about its properties.

- Australia Post, see "Australia Post, Japan Post, KIX, USPS IMb" on page 254
- "Aztec Code" on page 203
- "Codabar" on page 205
- Code 11, see "Code 11, Code 93, Industrial 2 of 5, Interleaved 2 of 5, Matrix 2 of 5" on page 232
- "Code 39" on page 208

- "Code 39 extended" on page 210
- Code 93, see "Code 11, Code 93, Industrial 2 of 5, Interleaved 2 of 5, Matrix 2 of 5" on page 232
- "Code 93 extended" on page 214
- "Code 128" on page 215
- "Data Matrix" on page 217
- EAN-**8**, EAN-**13**, see "UPC-A, UPC-E, EAN-8, EAN-13" on page 252
- "GS1 DataMatrix" on page 222
- "GS1-128" on page 224
- Industrial 2 of 5, see "Code 11, Code 93, Industrial 2 of 5, Interleaved 2 of 5, Matrix 2 of 5" on page 232
- Interleaved 2 of 5, see "Code 11, Code 93, Industrial 2 of 5, Interleaved 2 of 5, Matrix 2 of 5" on page 232
- Japan Post, see "Australia Post, Japan Post, KIX, USPS IMb" on page 254
- KIX, see "Australia Post, Japan Post, KIX, USPS IMb" on page 254
- Matrix 2 of 5, see "Code 11, Code 93, Industrial 2 of 5, Interleaved 2 of 5, Matrix 2 of 5" on page 232
- "MaxiCode" on page 234
- "MSI" on page 235
- "PDF417" on page 237
- "POSTNET" on page 239
- "QR Code" on page 240
- "Royal Mail 2D Mailmark" on page 243
- "Royal Mail 4-State (CBC)" on page 245
- Royal Mail 4-State Mailmark **C**, Royal Mail 4-State Mailmark **L**, see "Royal Mail 4-State Mailmark C, Royal Mail 4-State Mailmark L" on page 248
- "Swiss QR Code" on page 250
- UPC-**A**, UPC-**E**, see "UPC-A, UPC-E, EAN-8, EAN-13" on page 252
- USPS IMb, see "Australia Post, Japan Post, KIX, USPS IMb" on page 254
- "USPS IMpb" on page 255

Australia Post, Japan Post, KIX, USPS IMb

Australia Post, **Japan Post**, **KIX** and **USPS IMb** are some of the types of barcodes that can be added to a template; see "Barcode" on page 197.

In PrintShop Mail Connect versions prior to 2019.2 the USPS IMb barcode was called "OneCode".

The barcode can be added either using the Barcode toolbar button or through selecting **Insert > Barcode** on the menu; see "Adding a Barcode" on page 198.

Initially the barcode will have the barcode type's default properties. To change those properties, such as the scale and color, open the Barcode properties dialog: right-click the barcode (on the **Design** tab in the Workspace) and select the **Barcode...** option in the context menu.

Note that barcodes of these types don't process tilde characters (~) in the data as special characters.

Barcode properties

This topic lists the properties of the barcode types Australia Post, Japan Post, KIX and UPS IMb. For the properties of other barcode types, see "Barcode type and properties" on page 200.

Height, width and spacing

The height, width and spacing of the barcode are all measured in pixels (38 dpi).

- **Bar height**: The height of the (shorter) bars.
- **Extended bar height**: The total height of the extended bars.
- **Bar width**: The width of the bars.
- **Spacing**: The distance between the bars.

Scale

Defines if and how the rendered barcode is scaled in relation to the parent element:

- **None**: The barcode is rendered based on the module width.
- **Proportional**: The barcode is stretched up to where it fits either the width or height of the parent box, whichever requires the less stretching.

Color

The **Color** group allows you to choose a different **Barcode** color (instead of black) and **Background** color (instead of white), by typing a hexadecimal color value (see for example w3school's color picker).

When black is a composite of CMYK or RGB values, it may incur a color click charge on some printers. Check the **Output in Grayscale** option to make sure that pure black is used.

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is usually of higher quality, but may take longer to generate and is not compatible with Email output.
- **PNG**: Binary rasterized format. This has a slightly lower quality than SVG but is usually generated faster and will display properly in Email output.

Aztec Code

Aztec Code is one of the types of barcodes that can be added to a template; see "Barcode" on page 197.

The barcode can be added either using the Barcode toolbar button or through selecting **Insert > Barcode** on the menu; see "Adding a Barcode" on page 198.

Initially the barcode will have the barcode type's default properties. To change those properties, such as the scale and color, open the Barcode properties dialog: right-click the barcode (on the **Design** tab in the Workspace) and select the **Barcode...** option in the context menu.

Barcode properties

This topic lists the properties of the Aztec Code barcode type. For the properties of other barcode types, see "Barcode type and properties" on page 200.

Module size

Enter the size of the square modules in pixels.

Configuration type

Use the drop-down to select the format type used when creating the barcode: only full range format, only compact formats, or any format.

Preferred configuration

Use the drop-down to select the preferred format for the barcode. Note that the barcode generator may choose a different format if the data cannot be represented by the preferred format.

Encoding

Use the drop-down to select the encoding type:

- **Normal** can encode any character but is not very efficient for encoding binary values (above 128).
- **Binary** is to be used only if the data contains many bytes/characters above 128.

Error Correction Level

This option reserves a percentage of the symbol capacity for error correction. The recommended percentage for this type of barcode is 23.

Rune

When set to a value between 0 and 255, an Aztec Rune corresponding to the selected value is created. Set the Rune to -1 to disable this feature.

Scale

Defines if and how the rendered barcode is scaled in relation to the parent element:

- **None**: The barcode is rendered based on the module width.
- **Proportional**: The barcode is stretched up to where it fits either the width or height of the parent box, whichever requires the less stretching.

Tilde processing

Check this option to process tilde (~) characters in the data as special characters. (See the Java4less Barcodes Guide to learn what the tilde character can be used for.)
The tilde is expected to be followed by the 'd' character and 3 digits representing an ASCII

character: **~dNNN** . For example, ~d013 represents a carriage return.
Note that with this option checked, any tilde that needs to be included in the output must be escaped by adding another tilde: ~~.

Color

The **Color** group allows you to choose a different **Barcode** color (instead of black) and **Background** color (instead of white), by typing a hexadecimal color value (see for example w3school's color picker).

When black is a composite of CMYK or RGB values, it may incur a color click charge on some printers. Check the **Output in Grayscale** option to make sure that pure black is used.

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is usually of higher quality, but may take longer to generate and is not compatible with Email output.
- **PNG**: Binary rasterized format. This has a slightly lower quality than SVG but is usually generated faster and will display properly in Email output.

Codabar

Codabar is one of the barcode types that can be added to a template.

The barcode can be added either using the Barcode toolbar button or through selecting **Insert > Barcode** on the menu; see "Adding a Barcode" on page 198.

Initially the barcode will have the barcode type's default properties. To change those properties, such as the scale and color, open the Barcode properties dialog: right-click the barcode (on the **Design** tab in the Workspace) and select the **Barcode...** option in the context menu.

Note that this type of barcode processes tilde characters (~) in the data as special characters. See the Java4less Barcodes Guide to learn what the tilde character can be used for.

Barcode properties

This topic lists the properties of the Codabar barcode. For the properties of other barcode types, see "Barcode type and properties" on page 200.

Module width

Specifies the width of the narrow bars in centimeters. Changing this value to a higher value will make the barcode bigger when Scale is set to None.

Start Char and Stop Char

Use the drop-down to select the start and stop character for the barcode, which defines the encoding mode. Available characters are A, B, C.

Scale

Defines if and how the rendered barcode is scaled in relation to the parent element:

- **None**: The barcode is rendered based on the module width.
- **Fit to box**: The barcode is stretched to fit the parent box in both width and height.

Add Checksum

When checked, PrintShop Mail Connect will calculate a Checksum character and add that to the result of the Barcode script. If the value to be encoded is longer than 10 digits, a second check character will be calculated.

Human Readable Message

When this option is checked, PrintShop Mail Connect shows a human readable text below or above the barcode, as defined using the Text Position, using the specified font and font size. The font size is given in points (pt).

Color

The **Color** group allows you to choose a different **Barcode** color (instead of black) and **Background** color (instead of white), by typing a hexadecimal color value (see for example w3school's color picker).

When black is a composite of CMYK or RGB values, it may incur a color click charge on some printers. Check the **Output in Grayscale** option to make sure that pure black is used.

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is usually of higher quality, but may take longer to generate and is not compatible with Email output.
- **PNG**: Binary rasterized format. This has a slightly lower quality than SVG but is usually generated faster and will display properly in Email output.

Code 11, Code 93, Industrial 2 of 5, Interleaved 2 of 5, Matrix 2 of 5

Code 11, Code 93, Industrial 2 of 5, Interleaved 2 of 5, and Matrix 2 of 5 are a few of the barcode types that can be added to a template.

The barcode can be added either using the Barcode toolbar button or through selecting **Insert > Barcode** on the menu; see "Adding a Barcode" on page 198.

Initially the barcode will have the barcode type's default properties. To change those properties, such as the scale and color, open the Barcode properties dialog: right-click the barcode (on the **Design** tab in the Workspace) and select the **Barcode...** option in the context menu.

Note that barcodes of these types process tilde characters (~) in the data as special characters. See the Java4less Barcodes Guide to learn what the tilde character can be used for.

Barcode properties

This topic lists the properties of the following barcode types :

- Code 11
- Code 93
- Industrial 2 of 5
- Interleaved 2 of 5
- Matrix 2 of 5

For the properties of other barcode types, see "Barcode type and properties" on page 200.

Module width

Specifies the width of the narrow bars in centimeters. Changing this value to a higher value will make the barcode bigger when Scale is set to None.

Scale

Defines if and how the rendered barcode is scaled in relation to the parent element:

- **None**: The barcode is rendered based on the module width.
- **Fit to box**: The barcode is stretched to fit the parent box in both width and height.

Add Checksum

When checked, PrintShop Mail Connect will calculate a Checksum character and add that to the result of the Barcode script. If the value to be encoded is longer than 10 digits, a second check character will be calculated.

Human Readable Message

When this option is checked, PrintShop Mail Connect shows a human readable text below or above the barcode, as defined using the Text Position, using the specified font and font size. The font size is given in points (pt).

Color

The **Color** group allows you to choose a different **Barcode** color (instead of black) and **Background** color (instead of white), by typing a hexadecimal color value (see for example w3school's color picker).

When black is a composite of CMYK or RGB values, it may incur a color click charge on some printers. Check the **Output in Grayscale** option to make sure that pure black is used.

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is usually of higher quality, but may take longer to generate and is not compatible with Email output.
- **PNG**: Binary rasterized format. This has a slightly lower quality than SVG but is usually generated faster and will display properly in Email output.

Code 39

Code 39 is one of the barcode types that can be added to a template.

The barcode can be added either using the Barcode toolbar button or through selecting **Insert > Barcode** on the menu; see "Adding a Barcode" on page 198.

Initially the barcode will have the barcode type's default properties. To change those properties, such as the scale and color, open the Barcode properties dialog: right-click the barcode (on the **Design** tab in the Workspace) and select the **Barcode...** option in the context menu.

Note that barcodes of this type process tilde characters (~) in the data as special characters. See the Java4less Barcodes Guide to learn what the tilde character can be used for.

Barcode properties

This topic lists the properties of the barcode types Code 39 and Code 39 extended. For the properties of other barcode types, see "Barcode type and properties" on page 200.

Module width

Specifies the width of the narrow bars in centimeters. Changing this value to a higher value will make the barcode bigger when Scale is set to None.

Inter Character Gap

Two adjacent characters are separated by an inter-character gap. A value of 1 means that the separator will have the same length as the width of the narrow bars (in centimeters).

Scale

Defines if and how the rendered barcode is scaled in relation to the parent element:

- **None**: The barcode is rendered based on the module width.
- **Fit to box**: The barcode is stretched to fit the parent box in both width and height.

Add Checksum

When checked, PrintShop Mail Connect will calculate a Checksum character and add that to the result of the Barcode script. If the value to be encoded is longer than 10 digits, a second check character will be calculated.

Human Readable Message

When this option is checked, PrintShop Mail Connect shows a human readable text below or above the barcode, as defined using the Text Position, using the specified font and font size. The font size is given in points (pt).

Color

The **Color** group allows you to choose a different **Barcode** color (instead of black) and **Background** color (instead of white), by typing a hexadecimal color value (see for example w3school's color picker).

When black is a composite of CMYK or RGB values, it may incur a color click charge on some printers. Check the **Output in Grayscale** option to make sure that pure black is used.

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is usually of higher quality, but may take longer to generate and is not compatible with Email output.
- **PNG**: Binary rasterized format. This has a slightly lower quality than SVG but is usually generated faster and will display properly in Email output.

Code 39 extended

Code 39 extended is one of the barcode types that can be added to a template.

The barcode can be added either using the Barcode toolbar button or through selecting **Insert > Barcode** on the menu; see "Adding a Barcode" on page 198.

Initially the barcode will have the barcode type's default properties. To change those properties, such as the scale and color, open the Barcode properties dialog: right-click the barcode (on the **Design** tab in the Workspace) and select the **Barcode...** option in the context menu.

Barcode properties

This topic lists the properties of the Code 39 extended barcode type. For the properties of other barcode types, see "Barcode type and properties" on page 200.

Module width

Specifies the width of the narrow bars in centimeters. Changing this value to a higher value will make the barcode bigger when Scale is set to None.

Inter Character Gap

Two adjacent characters are separated by an inter-character gap. A value of 1 means that the separator will have the same length as the width of the narrow bars (in centimeters).

Scale

Defines if and how the rendered barcode is scaled in relation to the parent element:

- **None**: The barcode is rendered based on the module width.
- **Fit to box**: The barcode is stretched to fit the parent box in both width and height.

Tilde processing

Check this option to process tilde (~) characters in the data as special characters. (See the Java4less Barcodes Guide to learn what the tilde character can be used for.)
The tilde is expected to be followed by the 'd' character and 3 digits representing an ASCII character: **~dNNN** . For example, ~d013 represents a carriage return.
Note that with this option checked, any tilde that needs to be included in the output must be escaped by adding another tilde: ~~.

Add Checksum

When checked, PrintShop Mail Connect will calculate a Checksum character and add that to the result of the Barcode script. If the value to be encoded is longer than 10 digits, a second check character will be calculated.

Tilde processing

Check this option to process tilde (~) characters in the data as special characters. (See the Java4less Barcodes Guide to learn what the tilde character can be used for.)
The tilde is expected to be followed by the 'd' character and 3 digits representing an ASCII character: **~dNNN** . For example, ~d013 represents a carriage return.
Note that with this option checked, any tilde that needs to be included in the output must be escaped by adding another tilde: ~~.

Human Readable Message

When this option is checked, PrintShop Mail Connect shows a human readable text below or above the barcode, as defined using the Text Position, using the specified font and font size. The font size is given in points (pt).

Color

The **Color** group allows you to choose a different **Barcode** color (instead of black) and **Background** color (instead of white), by typing a hexadecimal color value (see for example w3school's color picker).

When black is a composite of CMYK or RGB values, it may incur a color click charge on some printers. Check the **Output in Grayscale** option to make sure that pure black is used.

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is usually of higher quality, but may take longer to generate and is not compatible with Email output.
- **PNG**: Binary rasterized format. This has a slightly lower quality than SVG but is usually generated faster and will display properly in Email output.

Code 11, Code 93, Industrial 2 of 5, Interleaved 2 of 5, Matrix 2 of 5

Code 11, Code 93, Industrial 2 of 5, Interleaved 2 of 5, and Matrix 2 of 5 are a few of the barcode types that can be added to a template.

The barcode can be added either using the Barcode toolbar button or through selecting **Insert > Barcode** on the menu; see "Adding a Barcode" on page 198.

Initially the barcode will have the barcode type's default properties. To change those properties, such as the scale and color, open the Barcode properties dialog: right-click the barcode (on the **Design** tab in the Workspace) and select the **Barcode...** option in the context menu.

Note that barcodes of these types process tilde characters (~) in the data as special characters. See the Java4less Barcodes Guide to learn what the tilde character can be used for.

Barcode properties

This topic lists the properties of the following barcode types :

- Code 11
- Code 93
- Industrial 2 of 5

- Interleaved 2 of 5
- Matrix 2 of 5

For the properties of other barcode types, see "Barcode type and properties" on page 200.

Module width

Specifies the width of the narrow bars in centimeters. Changing this value to a higher value will make the barcode bigger when Scale is set to None.

Scale

Defines if and how the rendered barcode is scaled in relation to the parent element:

- **None**: The barcode is rendered based on the module width.
- **Fit to box**: The barcode is stretched to fit the parent box in both width and height.

Add Checksum

When checked, PrintShop Mail Connect will calculate a Checksum character and add that to the result of the Barcode script. If the value to be encoded is longer than 10 digits, a second check character will be calculated.

Human Readable Message

When this option is checked, PrintShop Mail Connect shows a human readable text below or above the barcode, as defined using the Text Position, using the specified font and font size. The font size is given in points (pt).

Color

The **Color** group allows you to choose a different **Barcode** color (instead of black) and **Background** color (instead of white), by typing a hexadecimal color value (see for example w3school's color picker).

When black is a composite of CMYK or RGB values, it may incur a color click charge on some printers. Check the **Output in Grayscale** option to make sure that pure black is used.

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is usually of higher quality, but may take longer to generate and is not compatible with Email output.
- **PNG**: Binary rasterized format. This has a slightly lower quality than SVG but is usually generated faster and will display properly in Email output.

Code 93 extended

Code 93 extended is one of the barcode types that can be added to a template.

The barcode can be added either using the Barcode toolbar button or through selecting **Insert > Barcode** on the menu; see "Adding a Barcode" on page 198.

Initially the barcode will have the barcode type's default properties. To change those properties, such as the scale and color, open the Barcode properties dialog: right-click the barcode (on the **Design** tab in the Workspace) and select the **Barcode...** option in the context menu.

Barcode properties

This topic lists the properties of the **Code 93 extended** barcode. For the properties of other barcode types, see "Barcode type and properties" on page 200.

Module width

Specifies the width of the narrow bars in centimeters. Changing this value to a higher value will make the barcode bigger when Scale is set to None.

Scale

Defines if and how the rendered barcode is scaled in relation to the parent element:

- **None**: The barcode is rendered based on the module width.
- **Fit to box**: The barcode is stretched to fit the parent box in both width and height.

Add Checksum

When checked, PrintShop Mail Connect will calculate a Checksum character and add that to the result of the Barcode script. If the value to be encoded is longer than 10 digits, a second check character will be calculated.

Tilde processing

Check this option to process tilde (~) characters in the data as special characters. (See the Java4less Barcodes Guide to learn what the tilde character can be used for.)
The tilde is expected to be followed by the 'd' character and 3 digits representing an ASCII character: **~dNNN** . For example, ~d013 represents a carriage return.
Note that with this option checked, any tilde that needs to be included in the output must be escaped by adding another tilde: ~~.

Human Readable Message

When this option is checked, PrintShop Mail Connect shows a human readable text below or above the barcode, as defined using the Text Position, using the specified font and font size. The font size is given in points (pt).

Color

The **Color** group allows you to choose a different **Barcode** color (instead of black) and **Background** color (instead of white), by typing a hexadecimal color value (see for example w3school's color picker).

When black is a composite of CMYK or RGB values, it may incur a color click charge on some printers. Check the **Output in Grayscale** option to make sure that pure black is used.

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is usually of higher quality, but may take longer to generate and is not compatible with Email output.
- **PNG**: Binary rasterized format. This has a slightly lower quality than SVG but is usually generated faster and will display properly in Email output.

Code 128

Code 128 is one of the types of barcodes that can be added to a template; see "Barcode" on page 197.

The barcode can be added either using the Barcode toolbar button or through selecting **Insert > Barcode** on the menu; see "Adding a Barcode" on page 198.

Initially the barcode will have the barcode type's default properties. To change those properties, such as the scale and color, open the Barcode properties dialog: right-click the barcode (on the **Design** tab in the Workspace) and select the **Barcode...** option in the context menu.

Barcode properties

This topic lists the properties of the barcode type Code 128. For the properties of other barcode types, see "Barcode type and properties" on page 200.

Module width

Specifies the width of the narrow bars in centimeters. Changing this value to a higher value will make the barcode bigger when Scale is set to None.

Code set

Set of characters to be used:

- **A**: ASCII characters 00 to 95 (0–9, A–Z and control codes), special characters, and FNC 1–4
- **B**: ASCII characters 32 to 127 (0–9, A–Z, a–z), special characters, and FNC 1–4
- **C**: 00–99 (encodes each two digits with one code) and FNC 1

In **Auto** mode, the barcode generator will automatically select the correct encoding mode (set A, B or C) according to the input data.

If the barcode starts as one subset and then switches to another, set **Code set** to whatever the starting set is (not Auto). Enable the **Tilde processing** option (see below). At the position where the subset switches, add the following to the input data:

- ~d199 to switch to subset C
- ~d200 to switch to subset B
- ~d201 to switch to subset A

For example, if a barcode in the format AA123456789 must start as subset B, and then switch to subset C after 1, set the barcode to be: AA1**~d199**23456789.

Scale

Defines if and how the rendered barcode is scaled in relation to the parent element:

- **None**: The barcode is rendered based on the module width.
- **Fit to box**: The barcode is stretched to fit the parent box in both width and height.

Tilde processing

Check this option to process tilde (~) characters in the data as special characters. (See the Java4less Barcodes Guide to learn what the tilde character can be used for.)
The tilde is expected to be followed by the 'd' character and 3 digits representing an ASCII character: **~dNNN** . For example, ~d013 represents a carriage return.
Note that with this option checked, any tilde that needs to be included in the output must be escaped by adding another tilde: ~~.

Human Readable Message

When this option is checked, PrintShop Mail Connect shows a human readable text below or above the barcode, as defined using the Text Position, using the specified font and font size. The font size is given in points (pt).

Color

The **Color** group allows you to choose a different **Barcode** color (instead of black) and **Background** color (instead of white), by typing a hexadecimal color value (see for example w3school's color picker).

When black is a composite of CMYK or RGB values, it may incur a color click charge on some printers. Check the **Output in Grayscale** option to make sure that pure black is used.

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is usually of higher quality, but may take longer to generate and is not compatible with Email output.
- **PNG**: Binary rasterized format. This has a slightly lower quality than SVG but is usually generated faster and will display properly in Email output.

Data Matrix

Data Matrix is one of the types of barcodes that can be added to a template; see "Barcode" on page 197.

The barcode can be added either using the Barcode toolbar button or through selecting **Insert > Barcode** on the menu; see "Adding a Barcode" on page 198.

Initially the barcode will have the barcode type's default properties. To change those properties, such as the scale and color, open the Barcode properties dialog: right-click the barcode (on the **Design** tab in the Workspace) and select the **Barcode...** option in the context menu.

Data Matrix barcodes can be read even if they are partially damaged, because of the error correction codes added to them (according to the ECC200 standard).

Barcode properties

This topic lists the properties of the Data Matrix barcode. For the properties of other barcode types, see "Barcode type and properties" on page 200.

> **Note**
>
> **Tilde** characters in the data are processed as special characters (see the Java4less Barcodes Guide to learn what the tilde character can be used for).
> Any tilde that needs to be included in the output must be escaped by adding another tilde (either "~~", or "7E7E" if the Hex input option is enabled). This can be done with a `replace()` call in the Barcode script, after expanding the script.

Hex Input

For optimized mailings, German Post requires the supplied data for the Data Matrix barcode to be hexadecimal input.
Check this option if your input data is a hexadecimal code. The incoming data will be interpreted as hexadecimal input and decoded to ASCII before passing the string to the Barcode library.

Dots Per Pixels

Type the number of dots per pixel. To optimize barcode quality a Data Matrix symbol should not be printed with dots smaller than 4 pixels.

Encoding

The data represented in the symbol can be compressed using of the following algorithms.

- **ASCII** is used to encode data that mainly contains ascii characters (0-127)
- **C40** is used to encode data that mainly contains numbers and uppercase characters.
- **Text** is used to encode data that mainly contains numbers and lowercase
- **Base256** is used to encode 8 bit values
- **Auto Detect** automatically detects the data content and encodes using the most appropriate method.
- **None** does not use any encoding.

Preferred format

Use the drop-down to select the size of the Data Matrix.

Scale

Defines if and how the rendered barcode is scaled in relation to the parent element:

- **None**: The barcode is rendered based on the module width.
- **Proportional**: The barcode is stretched up to where it fits either the width or height of the parent box, whichever requires the less stretching.

Tilde processing

Check this option to process tilde (~) characters in the data as special characters. (See the Java4less Barcodes Guide to learn what the tilde character can be used for.)
The tilde is expected to be followed by the 'd' character and 3 digits representing an ASCII character: **~dNNN** . For example, ~d013 represents a carriage return.
Note that with this option checked, any tilde that needs to be included in the output must be escaped by adding another tilde: ~~.

Color

The **Color** group allows you to choose a different **Barcode** color (instead of black) and **Background** color (instead of white), by typing a hexadecimal color value (see for example w3school's color picker).

When black is a composite of CMYK or RGB values, it may incur a color click charge on some printers. Check the **Output in Grayscale** option to make sure that pure black is used.

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is usually of higher quality, but may take longer to generate and is not compatible with Email output.
- **PNG**: Binary rasterized format. This has a slightly lower quality than SVG but is usually generated faster and will display properly in Email output.

UPC-A, UPC-E, EAN-8, EAN-13

UPC-A, UPC-E, EAN-8 and EAN-13 are a few of the barcode types that can be added to a template.

The barcode can be added either using the Barcode toolbar button or through selecting **Insert > Barcode** on the menu; see "Adding a Barcode" on page 198.

Initially the barcode will have the barcode type's default properties. To change those properties, such as the scale and color, open the Barcode properties dialog: right-click the barcode (on the **Design** tab in the Workspace) and select the **Barcode...** option in the context menu.

Note that these barcode types process tilde characters in the data as special characters. See the Java4less Barcodes Guide to learn what the tilde character can be used for.

Barcode properties

This topic lists the properties of the barcode types UPC-A, UPC-E, EAN-8 and EAN-13. For the properties of other barcode types, see "Barcode type and properties" on page 200.

Module width

Specifies the width of the narrow bars in centimeters. Changing this value to a higher value will make the barcode bigger when Scale is set to None.

Show guardbars

Checking this option adds guardbars to the barcode. Guardbars are bars at the start, in the middle and at the end that help the barcode scanner to scan the barcode correctly.

Scale

Defines if and how the rendered barcode is scaled in relation to the parent element:

- **None**: The barcode is rendered based on the module width.
- **Fit to box**: The barcode is stretched to fit the parent box in both width and height.

Supplement

UPC-A, UPC-E, EAN-13, and EAN-8 may all include an additional barcode to the right of the main barcode.

- **Type**: The supplement type can be 2-digit (originally used to indicate the edition of a magazine or periodical) or 5-digit (used to indicate the suggested retail price for books). In case this option is set to None, and the data includes digits for the 2 or 5 supplement, the supplement data will be skipped and the additional barcode will not be rendered.

> **Note**
>
> When the chosen supplement type doesn't match the data, the supplement data will be skipped and the additional barcode will not be rendered.

- **Height Factor**: This is the relative height of the supplement's bars compared to the normal bars.
- **Space Before** : Defines the space between the main symbol and the supplement, in cm.

Human Readable Message

When this option is checked, PrintShop Mail Connect shows a human readable text below or above the barcode, as defined using the Text Position, using the specified font and font size. The font size is given in points (pt).

Color

The **Color** group allows you to choose a different **Barcode** color (instead of black) and **Background** color (instead of white), by typing a hexadecimal color value (see for example w3school's color picker).

When black is a composite of CMYK or RGB values, it may incur a color click charge on some printers. Check the **Output in Grayscale** option to make sure that pure black is used.

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is usually of higher quality, but may take longer to generate and is not compatible with Email output.
- **PNG**: Binary rasterized format. This has a slightly lower quality than SVG but is usually generated faster and will display properly in Email output.

GS1 DataMatrix

GS1 DataMatrix is one of the types of barcodes that can be added to a template; see "Barcode" on page 197.

The barcode can be added either using the Barcode toolbar button or through selecting **Insert > Barcode** on the menu; see "Adding a Barcode" on page 198.

Initially the barcode will have the barcode type's default properties. To change those properties, such as the scale and color, open the Barcode properties dialog: right-click the barcode (on the **Design** tab in the Workspace) and select the **Barcode...** option in the context menu.

About GS1 DataMatrix barcode data

In order to create a GS1 DataMatrix barcode, the barcode data must meet the following requirements.

- The barcode data must start with a leading FNC1 character. Either ~1 or ~d232.
- The GS1 Application Identifiers (AI) must be used for all data.
  The function code ~1 must be used as field separator for variable length AI elements.
- Only ASCII characters should be used.

Barcode properties

This topic lists the properties of the barcode type GS1 DataMatrix. For the properties of other barcode types, see "Barcode type and properties" on page 200.

Module width

Specifies the width of the narrow bars in centimeters. Changing this value to a higher value will make the barcode bigger when Scale is set to None.

Preferred format

Use the drop-down to select the size of the Data Matrix.

Scale

Defines if and how the rendered barcode is scaled in relation to the parent element:

- **None**: The barcode is rendered based on the module width.
- **Fit to box**: The barcode is stretched to fit the parent box in both width and height.

Add Checksum

When checked, PrintShop Mail Connect will calculate a Checksum character and add that to the result of the Barcode script. If the value to be encoded is longer than 10 digits, a second check character will be calculated.

Tilde processing

Check this option to process tilde (~) characters in the data as special characters. (See the Java4less Barcodes Guide to learn what the tilde character can be used for.)
The tilde is expected to be followed by the 'd' character and 3 digits representing an ASCII character: **~dNNN** . For example, ~d013 represents a carriage return.
Note that with this option checked, any tilde that needs to be included in the output must be escaped by adding another tilde: ~~.

> **Note**
>
> The following tilde codes are supported in GS1 DataMatrix barcodes:
>
> - ~1 = FNC1 character (for GS1 DataMatrix barcodes)
> - ~2 = Structure Append
> - ~3 = Reader programming
> - ~5 = Macro5
> - ~6 = Macro6
> - ~7 = ECI expressions

Human Readable Message

When this option is checked, PrintShop Mail Connect shows a human readable text below or above the barcode, as defined using the Text Position, using the specified font and font size. The font size is given in points (pt).

Color

The **Color** group allows you to choose a different **Barcode** color (instead of black) and **Background** color (instead of white), by typing a hexadecimal color value (see for example w3school's color picker).

When black is a composite of CMYK or RGB values, it may incur a color click charge on some printers. Check the **Output in Grayscale** option to make sure that pure black is used.

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is usually of higher quality, but may take longer to generate and is not compatible with Email output.
- **PNG**: Binary rasterized format. This has a slightly lower quality than SVG but is usually generated faster and will display properly in Email output.

GS1-128

GS1-128 is one of the types of barcodes that can be added to a template; see "Barcode" on page 197.

The barcode can be added either using the Barcode toolbar button or through selecting **Insert > Barcode** on the menu; see "Adding a Barcode" on page 198.

Initially the barcode will have the barcode type's default properties. To change those properties, such as the scale and color, open the Barcode properties dialog: right-click the barcode (on the **Design** tab in the Workspace) and select the **Barcode...** option in the context menu.

Barcode properties

This topic lists the properties of the barcode type GS1-128. For the properties of other barcode types, see "Barcode type and properties" on page 200.

Module width

Specifies the width of the narrow bars in centimeters. Changing this value to a higher value will make the barcode bigger when Scale is set to None.

Scale

Defines if and how the rendered barcode is scaled in relation to the parent element:

- **None**: The barcode is rendered based on the module width.
- **Fit to box**: The barcode is stretched to fit the parent box in both width and height.

Add Checksum

When checked, PrintShop Mail Connect will calculate a Checksum character and add that to the result of the Barcode script. If the value to be encoded is longer than 10 digits, a second check character will be calculated.

Tilde processing

Check this option to process tilde (~) characters in the data as special characters. (See the Java4less Barcodes Guide to learn what the tilde character can be used for.)
The tilde is expected to be followed by the 'd' character and 3 digits representing an ASCII character: **~dNNN** . For example, ~d013 represents a carriage return.
Note that with this option checked, any tilde that needs to be included in the output must be escaped by adding another tilde: ~~.

Human Readable Message

When this option is checked, PrintShop Mail Connect shows a human readable text below or above the barcode, as defined using the Text Position, using the specified font and font size. The font size is given in points (pt).

Color

The **Color** group allows you to choose a different **Barcode** color (instead of black) and **Background** color (instead of white), by typing a hexadecimal color value (see for example w3school's color picker).

When black is a composite of CMYK or RGB values, it may incur a color click charge on some printers. Check the **Output in Grayscale** option to make sure that pure black is used.

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is usually of higher quality, but may take longer to generate and is not compatible with Email output.
- **PNG**: Binary rasterized format. This has a slightly lower quality than SVG but is usually generated faster and will display properly in Email output.

Code 11, Code 93, Industrial 2 of 5, Interleaved 2 of 5, Matrix 2 of 5

Code 11, Code 93, Industrial 2 of 5, Interleaved 2 of 5, and Matrix 2 of 5 are a few of the barcode types that can be added to a template.

The barcode can be added either using the Barcode toolbar button or through selecting **Insert > Barcode** on the menu; see "Adding a Barcode" on page 198.

Initially the barcode will have the barcode type's default properties. To change those properties, such as the scale and color, open the Barcode properties dialog: right-click the barcode (on the **Design** tab in the Workspace) and select the **Barcode...** option in the context menu.

Note that barcodes of these types process tilde characters (~) in the data as special characters. See the Java4less Barcodes Guide to learn what the tilde character can be used for.

Barcode properties

This topic lists the properties of the following barcode types :

- Code 11
- Code 93
- Industrial 2 of 5
- Interleaved 2 of 5
- Matrix 2 of 5

For the properties of other barcode types, see "Barcode type and properties" on page 200.

Module width

Specifies the width of the narrow bars in centimeters. Changing this value to a higher value will make the barcode bigger when Scale is set to None.

Scale

Defines if and how the rendered barcode is scaled in relation to the parent element:

- **None**: The barcode is rendered based on the module width.
- **Fit to box**: The barcode is stretched to fit the parent box in both width and height.

Add Checksum

When checked, PrintShop Mail Connect will calculate a Checksum character and add that to the result of the Barcode script. If the value to be encoded is longer than 10 digits, a second check character will be calculated.

Human Readable Message

When this option is checked, PrintShop Mail Connect shows a human readable text below or above the barcode, as defined using the Text Position, using the specified font and font size. The font size is given in points (pt).

Color

The **Color** group allows you to choose a different **Barcode** color (instead of black) and **Background** color (instead of white), by typing a hexadecimal color value (see for example w3school's color picker).

When black is a composite of CMYK or RGB values, it may incur a color click charge on some printers. Check the **Output in Grayscale** option to make sure that pure black is used.

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is usually of higher quality, but may take longer to generate and is not compatible with Email output.
- **PNG**: Binary rasterized format. This has a slightly lower quality than SVG but is usually generated faster and will display properly in Email output.

Code 11, Code 93, Industrial 2 of 5, Interleaved 2 of 5, Matrix 2 of 5

Code 11, Code 93, Industrial 2 of 5, Interleaved 2 of 5, and Matrix 2 of 5 are a few of the barcode types that can be added to a template.

The barcode can be added either using the Barcode toolbar button or through selecting **Insert > Barcode** on the menu; see "Adding a Barcode" on page 198.

Initially the barcode will have the barcode type's default properties. To change those properties, such as the scale and color, open the Barcode properties dialog: right-click the barcode (on the **Design** tab in the Workspace) and select the **Barcode...** option in the context menu.

Note that barcodes of these types process tilde characters (~) in the data as special characters. See the Java4less Barcodes Guide to learn what the tilde character can be used for.

Barcode properties

This topic lists the properties of the following barcode types :

- Code 11
- Code 93
- Industrial 2 of 5
- Interleaved 2 of 5
- Matrix 2 of 5

For the properties of other barcode types, see "Barcode type and properties" on page 200.

Module width

Specifies the width of the narrow bars in centimeters. Changing this value to a higher value will make the barcode bigger when Scale is set to None.

Scale

Defines if and how the rendered barcode is scaled in relation to the parent element:

- **None**: The barcode is rendered based on the module width.
- **Fit to box**: The barcode is stretched to fit the parent box in both width and height.

Add Checksum

When checked, PrintShop Mail Connect will calculate a Checksum character and add that to the result of the Barcode script. If the value to be encoded is longer than 10 digits, a second check character will be calculated.

Human Readable Message

When this option is checked, PrintShop Mail Connect shows a human readable text below or above the barcode, as defined using the Text Position, using the specified font and font size. The font size is given in points (pt).

Color

The **Color** group allows you to choose a different **Barcode** color (instead of black) and **Background** color (instead of white), by typing a hexadecimal color value (see for example w3school's color picker).

When black is a composite of CMYK or RGB values, it may incur a color click charge on some printers. Check the **Output in Grayscale** option to make sure that pure black is used.

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is usually of higher quality, but may take longer to generate and is not compatible with Email output.
- **PNG**: Binary rasterized format. This has a slightly lower quality than SVG but is usually generated faster and will display properly in Email output.

Australia Post, Japan Post, KIX, USPS IMb

Australia Post, **Japan Post**, **KIX** and **USPS IMb** are some of the types of barcodes that can be added to a template; see "Barcode" on page 197.

In PrintShop Mail Connect versions prior to 2019.2 the USPS IMb barcode was called "OneCode".

The barcode can be added either using the Barcode toolbar button or through selecting **Insert > Barcode** on the menu; see "Adding a Barcode" on page 198.

Initially the barcode will have the barcode type's default properties. To change those properties, such as the scale and color, open the Barcode properties dialog: right-click the barcode (on the **Design** tab in the Workspace) and select the **Barcode...** option in the context menu.

Note that barcodes of these types don't process tilde characters (~) in the data as special characters.

Barcode properties

This topic lists the properties of the barcode types Australia Post, Japan Post, KIX and UPS IMb. For the properties of other barcode types, see "Barcode type and properties" on page 200.

Height, width and spacing

The height, width and spacing of the barcode are all measured in pixels (38 dpi).

- **Bar height**: The height of the (shorter) bars.
- **Extended bar height**: The total height of the extended bars.
- **Bar width**: The width of the bars.
- **Spacing**: The distance between the bars.

Scale

Defines if and how the rendered barcode is scaled in relation to the parent element:

- **None**: The barcode is rendered based on the module width.
- **Proportional**: The barcode is stretched up to where it fits either the width or height of the parent box, whichever requires the less stretching.

Color

The **Color** group allows you to choose a different **Barcode** color (instead of black) and **Background** color (instead of white), by typing a hexadecimal color value (see for example w3school's color picker).

When black is a composite of CMYK or RGB values, it may incur a color click charge on some printers. Check the **Output in Grayscale** option to make sure that pure black is used.

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is usually of higher quality, but may take longer to generate and is not compatible with Email output.

- **PNG**: Binary rasterized format. This has a slightly lower quality than SVG but is usually generated faster and will display properly in Email output.

Australia Post, Japan Post, KIX, USPS IMb

Australia Post, **Japan Post**, **KIX** and **USPS IMb** are some of the types of barcodes that can be added to a template; see "Barcode" on page 197.

In PrintShop Mail Connect versions prior to 2019.2 the USPS IMb barcode was called "OneCode".

The barcode can be added either using the Barcode toolbar button or through selecting **Insert > Barcode** on the menu; see "Adding a Barcode" on page 198.

Initially the barcode will have the barcode type's default properties. To change those properties, such as the scale and color, open the Barcode properties dialog: right-click the barcode (on the **Design** tab in the Workspace) and select the **Barcode...** option in the context menu.

Note that barcodes of these types don't process tilde characters (~) in the data as special characters.

Barcode properties

This topic lists the properties of the barcode types Australia Post, Japan Post, KIX and UPS IMb. For the properties of other barcode types, see "Barcode type and properties" on page 200.

Height, width and spacing

The height, width and spacing of the barcode are all measured in pixels (38 dpi).

- **Bar height**: The height of the (shorter) bars.
- **Extended bar height**: The total height of the extended bars.
- **Bar width**: The width of the bars.
- **Spacing**: The distance between the bars.

Scale

Defines if and how the rendered barcode is scaled in relation to the parent element:

- **None**: The barcode is rendered based on the module width.
- **Proportional**: The barcode is stretched up to where it fits either the width or height of the parent box, whichever requires the less stretching.

Color

The **Color** group allows you to choose a different **Barcode** color (instead of black) and **Background** color (instead of white), by typing a hexadecimal color value (see for example w3school's color picker).

When black is a composite of CMYK or RGB values, it may incur a color click charge on some printers. Check the **Output in Grayscale** option to make sure that pure black is used.

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is usually of higher quality, but may take longer to generate and is not compatible with Email output.
- **PNG**: Binary rasterized format. This has a slightly lower quality than SVG but is usually generated faster and will display properly in Email output.

Code 11, Code 93, Industrial 2 of 5, Interleaved 2 of 5, Matrix 2 of 5

Code 11, Code 93, Industrial 2 of 5, Interleaved 2 of 5, and Matrix 2 of 5 are a few of the barcode types that can be added to a template.

The barcode can be added either using the Barcode toolbar button or through selecting **Insert > Barcode** on the menu; see "Adding a Barcode" on page 198.

Initially the barcode will have the barcode type's default properties. To change those properties, such as the scale and color, open the Barcode properties dialog: right-click the barcode (on the **Design** tab in the Workspace) and select the **Barcode...** option in the context menu.

Note that barcodes of these types process tilde characters (~) in the data as special characters. See the Java4less Barcodes Guide to learn what the tilde character can be used for.

Barcode properties

This topic lists the properties of the following barcode types :

- Code 11
- Code 93
- Industrial 2 of 5
- Interleaved 2 of 5
- Matrix 2 of 5

For the properties of other barcode types, see "Barcode type and properties" on page 200.

Module width

Specifies the width of the narrow bars in centimeters. Changing this value to a higher value will make the barcode bigger when Scale is set to None.

Scale

Defines if and how the rendered barcode is scaled in relation to the parent element:

- **None**: The barcode is rendered based on the module width.
- **Fit to box**: The barcode is stretched to fit the parent box in both width and height.

Add Checksum

When checked, PrintShop Mail Connect will calculate a Checksum character and add that to the result of the Barcode script. If the value to be encoded is longer than 10 digits, a second check character will be calculated.

Human Readable Message

When this option is checked, PrintShop Mail Connect shows a human readable text below or above the barcode, as defined using the Text Position, using the specified font and font size. The font size is given in points (pt).

Color

The **Color** group allows you to choose a different **Barcode** color (instead of black) and **Background** color (instead of white), by typing a hexadecimal color value (see for example w3school's color picker).

When black is a composite of CMYK or RGB values, it may incur a color click charge on some printers. Check the **Output in Grayscale** option to make sure that pure black is used.

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is usually of higher quality, but may take longer to generate and is not compatible with Email output.
- **PNG**: Binary rasterized format. This has a slightly lower quality than SVG but is usually generated faster and will display properly in Email output.

MaxiCode

MaxiCode is one of the barcode types that can be added to a template; see "Barcode" on page 197.

The barcode can be added either using the Barcode toolbar button or through selecting **Insert > Barcode** on the menu; see "Adding a Barcode" on page 198.

Initially the barcode will have the barcode type's default properties. To change those properties, such as the scale and color, open the Barcode properties dialog: right-click the barcode (on the **Design** tab in the Workspace) and select the **Barcode...** option in the context menu.

Barcode properties

This topic lists the properties of the MaxiCode barcode. For the properties of other barcode types, see "Barcode type and properties" on page 200.

Resolution

Select the printer output definition for the barcode (200, 300, 400, 500 or 600 dpi).

Mode

PrintShop Mail Connect supports several modes; for an explanation of these modes see the MaxiCode page on Wikipedia.

Tilde processing

Check this option to process tilde (~) characters in the data as special characters. (See the Java4less Barcodes Guide to learn what the tilde character can be used for.)
The tilde is expected to be followed by the 'd' character and 3 digits representing an ASCII character: **~dNNN** . For example, ~d013 represents a carriage return.
Note that with this option checked, any tilde that needs to be included in the output must be escaped by adding another tilde: ~~.

Color

The **Color** group allows you to choose a different **Barcode** color (instead of black) and **Background** color (instead of white), by typing a hexadecimal color value (see for example w3school's color picker).

When black is a composite of CMYK or RGB values, it may incur a color click charge on some printers. Check the **Output in Grayscale** option to make sure that pure black is used.

MSI

MSI is one of the types of barcodes that can be added to a template; see "Barcode" on page 197.

The barcode can be added either using the Barcode toolbar button or through selecting **Insert > Barcode** on the menu; see "Adding a Barcode" on page 198.

Initially the barcode will have the barcode type's default properties. To change those properties, such as the scale and color, open the Barcode properties dialog: right-click the barcode (on the **Design** tab in the Workspace) and select the **Barcode...** option in the context menu.

Note that this type of barcode processes tilde characters (~) in the data as special characters. See the Java4less Barcodes Guide to learn what the tilde character can be used for.

Barcode properties

This topic lists the properties of the barcode type MSI. For the properties of other barcode types, see "Barcode type and properties" on page 200.

Module width

Specifies the width of the narrow bars in centimeters. Changing this value to a higher value will make the barcode bigger when Scale is set to None.

Scale

Defines if and how the rendered barcode is scaled in relation to the parent element:

- **None**: The barcode is rendered based on the module width.
- **Fit to box**: The barcode is stretched to fit the parent box in both width and height.

Add Checksum

When checked, PrintShop Mail Connect will calculate a Checksum character and add that to the result of the Barcode script. If the value to be encoded is longer than 10 digits, a second check character will be calculated.

Checksum Type

The Checksum type can be MSI10, MSI11, MSI1010 or MSI1110; see https://en.wikipedia.org/wiki/MSI_Barcode.

Human Readable Message

When this option is checked, PrintShop Mail Connect shows a human readable text below or above the barcode, as defined using the Text Position, using the specified font and font size. The font size is given in points (pt).

Color

The **Color** group allows you to choose a different **Barcode** color (instead of black) and **Background** color (instead of white), by typing a hexadecimal color value (see for example w3school's color picker).

When black is a composite of CMYK or RGB values, it may incur a color click charge on some printers. Check the **Output in Grayscale** option to make sure that pure black is used.

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is usually of higher quality, but may take longer to generate and is not compatible with Email output.
- **PNG**: Binary rasterized format. This has a slightly lower quality than SVG but is usually generated faster and will display properly in Email output.

PDF417

PDF417 is one of the types of barcodes that can be added to a template; see "Barcode" on page 197.

The barcode can be added either using the Barcode toolbar button or through selecting **Insert > Barcode** on the menu; see "Adding a Barcode" on page 198.

Initially the barcode will have the barcode type's default properties. To change those properties, such as the scale and color, open the Barcode properties dialog: right-click the barcode (on the **Design** tab in the Workspace) and select the **Barcode...** option in the context menu.

Barcode properties

This topic lists the properties of the barcode type PDF417. For the properties of other barcode types, see "Barcode type and properties" on page 200.

Mode

Use the drop-down to set the compaction mode:

- **Binary**: allows any byte value to be encoded
- **Text**: allows all printable ASCII characters to be encoded (values from 32 to 126 and some additional control characters)
- **Numeric**: a more efficient mode for encoding numeric data

Error Correction Level

Use the drop-down to select the built-in error correction method based on Reed-Solomon algorithms. The error correction level is adjustable between level 0 (just error detection) and level 8 (maximum error correction). Recommended error correction levels are between level 2 and 5, but the optimal value depends on the amount of data, printing quality of the PDF417 symbol and decoding capabilities of the scanner.

Nr. of Columns

The number of data columns can vary from 3 to 30.

Nr. of Rows

A PDF417 bar code can have anywhere from 3 to 90 rows.

Bar height

Defines the height of the bars for a single row measured in pixels drawn.

Compact

Check this option to use Compact PDF417 instead of the PDF417 barcode. This shortened form of the PDF417 barcode is useful where the space for the symbol is restricted.

Scale

Defines if and how the rendered barcode is scaled in relation to the parent element:

- **None**: The barcode is rendered based on the module width.
- **Fit to box**: The barcode is stretched to fit the parent box in both width and height.
- **Proportionally**: The barcode is stretched up to where it fits either the width or height of the parent box, whichever requires the less stretching.

Tilde processing

Check this option to process tilde (~) characters in the data as special characters. (See the Java4less Barcodes Guide to learn what the tilde character can be used for.)
The tilde is expected to be followed by the 'd' character and 3 digits representing an ASCII character: **~dNNN** . For example, ~d013 represents a carriage return.
Note that with this option checked, any tilde that needs to be included in the output must be escaped by adding another tilde: ~~.

Color

The **Color** group allows you to choose a different **Barcode** color (instead of black) and **Background** color (instead of white), by typing a hexadecimal color value (see for example w3school's color picker).

When black is a composite of CMYK or RGB values, it may incur a color click charge on some printers. Check the **Output in Grayscale** option to make sure that pure black is used.

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is usually of higher quality, but may take longer to generate and is not compatible with Email output.
- **PNG**: Binary rasterized format. This has a slightly lower quality than SVG but is usually generated faster and will display properly in Email output.

POSTNET

POSTNET is one of the barcode types that can be added to a template; see "Barcode" on page 197.

The barcode can be added either using the Barcode toolbar button or through selecting **Insert > Barcode** on the menu; see "Adding a Barcode" on page 198.

Initially the barcode will have the barcode type's default properties. To change those properties, such as the scale and color, open the Barcode properties dialog: right-click the barcode (on the **Design** tab in the Workspace) and select the **Barcode...** option in the context menu.

Note that this type of barcode processes tilde characters (~) in the data as special characters. See the Java4less Barcodes Guide to learn what the tilde character can be used for.

Barcode properties

This topic lists the properties of the barcode type POSTNET. For the properties of other barcode types, see "Barcode type and properties" on page 200.

Module width

Specifies the width of the narrow bars in centimeters. Changing this value to a higher value will make the barcode bigger when Scale is set to None.

Bar height

You can set the height (in cm) of the short bars and the tall bars in the Postnet barcode.

Scale

Defines if and how the rendered barcode is scaled in relation to the parent element:

- **None**: The barcode is rendered based on the module width.
- **Fit to box**: The barcode is stretched to fit the parent box in both width and height.
- **Proportionally**: The barcode is stretched up to where it fits either the width or height of the parent box, whichever requires the less stretching.

Human Readable Message

When this option is checked, PrintShop Mail Connect shows a human readable text below or above the barcode, as defined using the Text Position, using the specified font and font size. The font size is given in points (pt).

Color

The **Color** group allows you to choose a different **Barcode** color (instead of black) and **Background** color (instead of white), by typing a hexadecimal color value (see for example w3school's color picker).

When black is a composite of CMYK or RGB values, it may incur a color click charge on some printers. Check the **Output in Grayscale** option to make sure that pure black is used.

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is usually of higher quality, but may take longer to generate and is not compatible with Email output.
- **PNG**: Binary rasterized format. This has a slightly lower quality than SVG but is usually generated faster and will display properly in Email output.

QR Code

A QR Code is one of the types of barcodes that can be added to a template; see "Barcode" on page 197.

The barcode can be added either using the Barcode toolbar button or through selecting **Insert > Barcode** on the menu; see "Adding a Barcode" on page 198.

Initially the barcode will have the barcode type's default properties. To change those properties, such as the scale and color, open the Barcode properties dialog: right-click the barcode (on the **Design** tab in the Workspace) and select the **Barcode...** option in the context menu.

Instead of using the Script wizard (see "Adding a Barcode" on page 198) you could write your own script to get the data for the QR Code.

Barcode properties

This topic lists the properties of the QR barcode. For the properties of other barcode types, see "Barcode type and properties" on page 200.

Preferred version

There are 40 sizes of QR codes. Select the preferred version for the QR code.

Auto configure

When this option is checked, the barcode generator overwrites the selected Preferred version (the previous selection) and defines the barcode version based upon on the supplied data.

Module size

Enter the size of the square modules in pixels.

Encoding

This option defines the encoding of the barcode. When **Auto** is selected, the barcode generator determines the encoding based on the supplied string. The other options are:

- **Numeric**: 10 bits per 3 digits, with a maximum of 7089 numerical characters.
- **Alphanumeric**: 11 bits per 2 characters, with a maximum of 4296 alphanumerical characters.
- **Byte**: 8 bits per character, with a maximum of 2953 characters.
- **Kanji**: 13 bits per character, with a maximum of 1817 characters.

Extended Channel Interpretation (ECI)

This setting enables data using character sets other than the default set. Select **Latin-1**, **Latin-2**, **Shift JIS** or **UTF-8**, or select **None** to disable extended channel interpretation.

Correction level

Part of the robustness of QR codes in the physical environment is their ability to sustain 'damage' and continue to function even when a part of the QR code image is obscured, defaced or removed. A higher correction level duplicates data within the QR Code to that effect, making it larger.

FNC

Use the drop-down to either disable FNC or select a FNC option:

- **No**: No FNC option.
- **First**: This mode indicator identifies symbols encoding data formatted according to the UCC/EAN Application Identifiers.
- **Second**: This mode indicator identifies symbols formatted in accordance with specific industry or application specifications previously agreed with AIM International. You must then set a value for the Application Indicator property.

Application Indicator

Enter the desired Application Indicator value here.

Scale

Defines if and how the rendered barcode is scaled in relation to the parent element:

- **None**: The barcode is rendered based on the module width.
- **Proportional**: The barcode is stretched up to where it fits either the width or height of the parent box, whichever requires the less stretching.

Tilde processing

Check this option to process tilde (~) characters in the data as special characters. (See the Java4less Barcodes Guide to learn what the tilde character can be used for.)
The tilde is expected to be followed by the 'd' character and 3 digits representing an ASCII character: **~dNNN** . For example, ~d013 represents a carriage return.
Note that with this option checked, any tilde that needs to be included in the output must be escaped by adding another tilde: ~~.

Color

The **Color** group allows you to choose a different **Barcode** color (instead of black) and **Background** color (instead of white), by typing a hexadecimal color value (see for example w3school's color picker).

When black is a composite of CMYK or RGB values, it may incur a color click charge on some printers. Check the **Output in Grayscale** option to make sure that pure black is used.

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is usually of higher quality, but may take longer to generate and is not compatible with Email output.
- **PNG**: Binary rasterized format. This has a slightly lower quality than SVG but is usually generated faster and will display properly in Email output.

Barcode Data

QR Codes can have many different types of data, which determines how the code will be generated. On top of just straightforward data, special data structures are used to trigger actions on the device that reads them. This can include contact cards, phone numbers, URLs, emails, etc.

To learn more about the specifications of the different QR code types, see the ZXing Project barcode contents page.

Royal Mail 2D Mailmark

Royal Mail 2D Mailmark is one of the types of barcodes that can be added to a template; see "Barcode" on page 197.

The barcode can be added either using the Barcode toolbar button or through selecting **Insert > Barcode** on the menu; see "Adding a Barcode" on page 198.

Initially the barcode will have the barcode type's default properties. To change those properties, such as the scale and color, open the Barcode properties dialog: right-click the barcode (on the **Design** tab in the Workspace) and select the **Barcode...** option in the context menu.

Barcode properties

This topic lists the properties of the barcode type Royal Mail 2D Mailmark. For the properties of other barcode types, see "Barcode type and properties" on page 200.

Module width

The recommendation is to print these barcodes with a module size of 0.5 mm, which equates to 6 dots when printed at 300dpi. The maximum module size for printing is 0.7 mm.

Preferred version

Use the drop-down to select the size of the barcode, in a number of modules. The actual size of the barcode can be 12 mm x 12 mm up to 22.4 mm x 22.4 mm, depending on the preferred version and the module width.

Scale

Defines if and how the rendered barcode is scaled in relation to the parent element:

- **None**: The barcode is rendered based on the module width.
- **Proportional**: The barcode is stretched up to where it fits either the width or height of the parent box, whichever requires the less stretching.

Tilde processing

Check this option to process tilde (~) characters in the data as special characters. (See the Java4less Barcodes Guide to learn what the tilde character can be used for.)
The tilde is expected to be followed by the 'd' character and 3 digits representing an ASCII character: **~dNNN** . For example, ~d013 represents a carriage return.
Note that with this option checked, any tilde that needs to be included in the output must be escaped by adding another tilde: ~~.

Color

The **Color** group allows you to choose a different **Barcode** color (instead of black) and **Background** color (instead of white), by typing a hexadecimal color value (see for example w3school's color picker).

When black is a composite of CMYK or RGB values, it may incur a color click charge on some printers. Check the **Output in Grayscale** option to make sure that pure black is used.

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is usually of higher quality, but may take longer to generate and is not compatible with Email output.
- **PNG**: Binary rasterized format. This has a slightly lower quality than SVG but is usually generated faster and will display properly in Email output.

Royal Mail 4-State (CBC)

Royal Mail 4-State (CBC) is one of the types of barcodes that can be added to a template; see "Barcode" on page 197.
In PrintShop Mail Connect versions prior to 2019.2 this barcode was called "Royal Mail 4 State (RM4SCC)".

The barcode can be added either using the Barcode toolbar button or through selecting **Insert > Barcode** on the menu; see "Adding a Barcode" on page 198.

Initially the barcode will have the barcode type's default properties. To change those properties, such as the scale and color, open the Barcode properties dialog: right-click the barcode (on the **Design** tab in the Workspace) and select the **Barcode...** option in the context menu.

Note that this type of barcode doesn't process tilde characters (~) in the data as special characters.

Barcode properties

This topic lists the properties of the barcode type Royal Mail 4-State (CBC). For the properties of other barcode types, see "Barcode type and properties" on page 200.

Height, width and spacing

The height, width and spacing of the barcode are all measured in pixels (38 dpi).

- **Bar height**: the height of the (shorter) bars.
- **Extended bar height**: the total height of the extended bars.
- **Bar width**: the width of the bars.
- **Spacing**: the distance between the bars.

Scale

Defines if and how the rendered barcode is scaled in relation to the parent element:

- **None**: The barcode is rendered based on the module width.
- **Proportional**: The barcode is stretched up to where it fits either the width or height of the parent box, whichever requires the less stretching.

Add Checksum

When checked, PrintShop Mail Connect will calculate a Checksum character and add that to the result of the Barcode script. If the value to be encoded is longer than 10 digits, a second check character will be calculated.

Human Readable Message

When this option is checked, PrintShop Mail Connect shows a human readable text below or above the barcode, as defined using the Text Position, using the specified font and font size. The font size is given in points (pt).

Color

The **Color** group allows you to choose a different **Barcode** color (instead of black) and **Background** color (instead of white), by typing a hexadecimal color value (see for example w3school's color picker).

When black is a composite of CMYK or RGB values, it may incur a color click charge on some printers. Check the **Output in Grayscale** option to make sure that pure black is used.

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is usually of higher quality, but may take longer to generate and is not compatible with Email output.
- **PNG**: Binary rasterized format. This has a slightly lower quality than SVG but is usually generated faster and will display properly in Email output.

Royal Mail 4-State Mailmark C, Royal Mail 4-State Mailmark L

Royal Mail 4-State Mailmark C and Royal Mail 4-State Mailmark L are some of the types of barcodes that can be added to a template; see "Barcode" on page 197.

The barcode can be added either using the Barcode toolbar button or through selecting **Insert > Barcode** on the menu; see "Adding a Barcode" on page 198.

Initially the barcode will have the barcode type's default properties. To change those properties, such as the scale and color, open the Barcode properties dialog: right-click the barcode (on the **Design** tab in the Workspace) and select the **Barcode...** option in the context menu.

Barcode properties

This topic lists the properties of the barcode types Royal Mail 4-State Mailmark C and Royal Mail 4-State Mailmark L.
For the properties of other barcode types, see "Barcode type and properties" on page 200.

Height, width and spacing

The height, width and spacing of the barcode are all measured in pixels (38 dpi).

- **Bar height**: The height of the (shorter) bars.
- **Extended bar height**: The total height of the extended bars.
- **Bar width**: The width of the bars.
- **Spacing**: The distance between the bars.

Scale

Defines if and how the rendered barcode is scaled in relation to the parent element:

- **None**: The barcode is rendered based on the module width.
- **Proportional**: The barcode is stretched up to where it fits either the width or height of the parent box, whichever requires the less stretching.

Human Readable Message

When this option is checked, PrintShop Mail Connect shows a human readable text below or above the barcode, as defined using the Text Position, using the specified font and font size. The font size is given in points (pt).

Color

The **Color** group allows you to choose a different **Barcode** color (instead of black) and **Background** color (instead of white), by typing a hexadecimal color value (see for example w3school's color picker).

When black is a composite of CMYK or RGB values, it may incur a color click charge on some printers. Check the **Output in Grayscale** option to make sure that pure black is used.

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is usually of higher quality, but may take longer to generate and is not compatible with Email output.
- **PNG**: Binary rasterized format. This has a slightly lower quality than SVG but is usually generated faster and will display properly in Email output.

Royal Mail 4-State Mailmark C, Royal Mail 4-State Mailmark L

Royal Mail 4-State Mailmark C and Royal Mail 4-State Mailmark L are some of the types of barcodes that can be added to a template; see "Barcode" on page 197.

The barcode can be added either using the Barcode toolbar button or through selecting **Insert > Barcode** on the menu; see "Adding a Barcode" on page 198.

Initially the barcode will have the barcode type's default properties. To change those properties, such as the scale and color, open the Barcode properties dialog: right-click the barcode (on the **Design** tab in the Workspace) and select the **Barcode...** option in the context menu.

Barcode properties

This topic lists the properties of the barcode types Royal Mail 4-State Mailmark C and Royal Mail 4-State Mailmark L.
For the properties of other barcode types, see "Barcode type and properties" on page 200.

Height, width and spacing

The height, width and spacing of the barcode are all measured in pixels (38 dpi).

- **Bar height**: The height of the (shorter) bars.
- **Extended bar height**: The total height of the extended bars.
- **Bar width**: The width of the bars.
- **Spacing**: The distance between the bars.

Scale

Defines if and how the rendered barcode is scaled in relation to the parent element:

- **None**: The barcode is rendered based on the module width.
- **Proportional**: The barcode is stretched up to where it fits either the width or height of the parent box, whichever requires the less stretching.

Human Readable Message

When this option is checked, PrintShop Mail Connect shows a human readable text below or above the barcode, as defined using the Text Position, using the specified font and font size. The font size is given in points (pt).

Color

The **Color** group allows you to choose a different **Barcode** color (instead of black) and **Background** color (instead of white), by typing a hexadecimal color value (see for example w3school's color picker).

When black is a composite of CMYK or RGB values, it may incur a color click charge on some printers. Check the **Output in Grayscale** option to make sure that pure black is used.

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is usually of higher quality, but may take longer to generate and is not compatible with Email output.
- **PNG**: Binary rasterized format. This has a slightly lower quality than SVG but is usually generated faster and will display properly in Email output.

Swiss QR Code

The Swiss QR Code is a specific sub-set of QR barcode standards and is one of the several types of barcodes that can be added to a template; see "Barcode" on page 197.

The barcode can be added either using the Barcode toolbar button or through selecting **Insert > Barcode** on the menu; see "Adding a Barcode" on page 198.

Initially the barcode will have the barcode type's default properties. To change those properties, such as the scale and color, open the Barcode properties dialog: right-click the barcode (on the **Design** tab in the Workspace) and select the **Barcode...** option in the context menu.

Instead of using the Script wizard (see "Adding a Barcode" on page 198) you could write your own script to get the data for the QR Code.

Barcode properties

This topic lists the properties available for Swiss QR barcodes. For the properties of other barcode types, see "Barcode type and properties" on page 200.

Encoding

This option defines the encoding of the barcode. When **Auto** is selected, the barcode generator determines the encoding based on the supplied string. The other options are:

- **Numeric**: 10 bits per 3 digits, with a maximum of 7089 numerical characters.
- **Alphanumeric**: 11 bits per 2 characters, with a maximum of 4296 alphanumerical characters.
- **Byte**: 8 bits per character, with a maximum of 2953 characters.
- **Kanji**: 13 bits per character, with a maximum of 1817 characters.

FNC

Use the drop-down to either disable FNC or select a FNC option:

- **No**: No FNC option.
- **First**: This mode indicator identifies symbols encoding data formatted according to the UCC/EAN Application Identifiers.

- **Second**: This mode indicator identifies symbols formatted in accordance with specific industry or application specifications previously agreed with AIM International. You must then set a value for the Application Indicator property.

Tilde processing

Check this option to process tilde (~) characters in the data as special characters. (See the Java4less Barcodes Guide to learn what the tilde character can be used for.)
The tilde is expected to be followed by the 'd' character and 3 digits representing an ASCII character: **~dNNN** . For example, ~d013 represents a carriage return.
Note that with this option checked, any tilde that needs to be included in the output must be escaped by adding another tilde: ~~.

Color

The **Color** group allows you to choose a different **Barcode** color (instead of black) and **Background** color (instead of white), by typing a hexadecimal color value (see for example w3school's color picker).

When black is a composite of CMYK or RGB values, it may incur a color click charge on some printers. Check the **Output in Grayscale** option to make sure that pure black is used.

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is usually of higher quality, but may take longer to generate and is not compatible with Email output.
- **PNG**: Binary rasterized format. This has a slightly lower quality than SVG but is usually generated faster and will display properly in Email output.

Barcode Data

To learn more about the specifications of the Swiss QR code , please see the Swiss Payments Standard page, or download the barcode specification ig-qr-bill-en.pdf.

> **Note**
>
> The Swiss QR code standard dictates that the barcode dimensions must be 46 mm by 46

mm. Please do not change the barcode dimensions in the entry as this will invalidate the barcode.

UPC-A, UPC-E, EAN-8, EAN-13

UPC-A, UPC-E, EAN-8 and EAN-13 are a few of the barcode types that can be added to a template.

The barcode can be added either using the Barcode toolbar button or through selecting **Insert > Barcode** on the menu; see "Adding a Barcode" on page 198.

Initially the barcode will have the barcode type's default properties. To change those properties, such as the scale and color, open the Barcode properties dialog: right-click the barcode (on the **Design** tab in the Workspace) and select the **Barcode...** option in the context menu.

Note that these barcode types process tilde characters in the data as special characters. See the Java4less Barcodes Guide to learn what the tilde character can be used for.

Barcode properties

This topic lists the properties of the barcode types UPC-A, UPC-E, EAN-8 and EAN-13. For the properties of other barcode types, see "Barcode type and properties" on page 200.

Module width

Specifies the width of the narrow bars in centimeters. Changing this value to a higher value will make the barcode bigger when Scale is set to None.

Show guardbars

Checking this option adds guardbars to the barcode. Guardbars are bars at the start, in the middle and at the end that help the barcode scanner to scan the barcode correctly.

Scale

Defines if and how the rendered barcode is scaled in relation to the parent element:

- **None**: The barcode is rendered based on the module width.
- **Fit to box**: The barcode is stretched to fit the parent box in both width and height.

Supplement

UPC-A, UPC-E, EAN-13, and EAN-8 may all include an additional barcode to the right of the main barcode.

- **Type**: The supplement type can be 2-digit (originally used to indicate the edition of a magazine or periodical) or 5-digit (used to indicate the suggested retail price for books). In case this option is set to None, and the data includes digits for the 2 or 5 supplement, the supplement data will be skipped and the additional barcode will not be rendered.

  > **Note**
  >
  > When the chosen supplement type doesn't match the data, the supplement data will be skipped and the additional barcode will not be rendered.

- **Height Factor**: This is the relative height of the supplement's bars compared to the normal bars.
- **Space Before** : Defines the space between the main symbol and the supplement, in cm.

Human Readable Message

When this option is checked, PrintShop Mail Connect shows a human readable text below or above the barcode, as defined using the Text Position, using the specified font and font size. The font size is given in points (pt).

Color

The **Color** group allows you to choose a different **Barcode** color (instead of black) and **Background** color (instead of white), by typing a hexadecimal color value (see for example w3school's color picker).

When black is a composite of CMYK or RGB values, it may incur a color click charge on some printers. Check the **Output in Grayscale** option to make sure that pure black is used.

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is usually of higher quality, but may take longer to generate and is not compatible with Email output.
- **PNG**: Binary rasterized format. This has a slightly lower quality than SVG but is usually generated faster and will display properly in Email output.

Australia Post, Japan Post, KIX, USPS IMb

Australia Post, **Japan Post**, **KIX** and **USPS IMb** are some of the types of barcodes that can be added to a template; see "Barcode" on page 197.

In PrintShop Mail Connect versions prior to 2019.2 the USPS IMb barcode was called "OneCode".

The barcode can be added either using the Barcode toolbar button or through selecting **Insert > Barcode** on the menu; see "Adding a Barcode" on page 198.

Initially the barcode will have the barcode type's default properties. To change those properties, such as the scale and color, open the Barcode properties dialog: right-click the barcode (on the **Design** tab in the Workspace) and select the **Barcode...** option in the context menu.

Note that barcodes of these types don't process tilde characters (~) in the data as special characters.

Barcode properties

This topic lists the properties of the barcode types Australia Post, Japan Post, KIX and UPS IMb. For the properties of other barcode types, see "Barcode type and properties" on page 200.

Height, width and spacing

The height, width and spacing of the barcode are all measured in pixels (38 dpi).

- **Bar height**: The height of the (shorter) bars.
- **Extended bar height**: The total height of the extended bars.
- **Bar width**: The width of the bars.
- **Spacing**: The distance between the bars.

Scale

Defines if and how the rendered barcode is scaled in relation to the parent element:

- **None**: The barcode is rendered based on the module width.
- **Proportional**: The barcode is stretched up to where it fits either the width or height of the parent box, whichever requires the less stretching.

Color

The **Color** group allows you to choose a different **Barcode** color (instead of black) and **Background** color (instead of white), by typing a hexadecimal color value (see for example w3school's color picker).

When black is a composite of CMYK or RGB values, it may incur a color click charge on some printers. Check the **Output in Grayscale** option to make sure that pure black is used.

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is usually of higher quality, but may take longer to generate and is not compatible with Email output.
- **PNG**: Binary rasterized format. This has a slightly lower quality than SVG but is usually generated faster and will display properly in Email output.

USPS IMpb

USPS IMpb is one of the barcode types that can be added to a template; see "Barcode" on page 197.
In PrintShop Mail Connect versions prior to 2019.2 this barcode was called "IMPB".

The barcode can be added either using the Barcode toolbar button or through selecting **Insert > Barcode** on the menu; see "Adding a Barcode" on page 198.

Initially the barcode will have the barcode type's default properties. To change those properties, such as the scale and color, open the Barcode properties dialog: right-click the barcode (on the **Design** tab in the Workspace) and select the **Barcode...** option in the context menu.

Note that this type of barcode processes tilde characters (~) in the data as special characters. See the Java4less Barcodes Guide to learn what the tilde character can be used for.

Barcode properties

This topic lists the properties of the USPS IMpb barcode type. For the properties of other barcode types, see "Barcode type and properties" on page 200.

Color

The **Color** group allows you to choose a different **Barcode** color (instead of black) and **Background** color (instead of white), by typing a hexadecimal color value (see for example w3school's color picker).

When black is a composite of CMYK or RGB values, it may incur a color click charge on some printers. Check the **Output in Grayscale** option to make sure that pure black is used.

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is usually of higher quality, but may take longer to generate and is not compatible with Email output.
- **PNG**: Binary rasterized format. This has a slightly lower quality than SVG but is usually generated faster and will display properly in Email output.

Boxes

Boxes are elements that are used to surround other elements, either to style them, to find them, or to place them in specific locations.

> **Tip**
>
> Wrapping elements in a box (or in a semantic HTML element) makes it easier to target them in a script or in a style sheet. Place the cursor in the element or select multiple elements. Then, on the menu, click **Insert > Wrap in Box**. You can now use the wrapper element as a script's or style's `selector`; see "Using the Text Script Wizard" on page 353 and "Styling and formatting" on page 294.

Tip

With the **Copy fit** feature, text can automatically be scaled to the available space in a Box or Div. See "Copy Fit" on page 311.

Positioned Box

A Positioned Box is one that can be freely moved around the page and does not depend on the position of other elements. A positioned box is actually a element that has an **absolute position**; in other words, it has its CSS property `position` set to `absolute` (see also: "Using the CSS position property" on page 314).

Positioned Boxes are suitable for use in Print templates only.

Adding a Positioned Box

To insert a Positioned Box, use the ▭ icon on the toolbar.

Moving and resizing a Positioned Box

Positioned Boxes can be moved by **dragging** the borders, and resized using the handles on the sides and the corners. Pressing any arrow key moves the box by 1 pixel in the direction of that key.
Alternatively the size and position can be set on the **Attributes** pane. Note that the size and offset values will be displayed in the default print units as defined in the preferences.

- To move or resize multiple Positioned Boxes at the same time, hold the **Ctrl** key while selecting them. You could either select them in the Design view (the main editor) or in the Outline pane.
- Hold the **Shift** key while resizing a Positioned Box to ensures that the box retains its aspect ratio.
- Hold the **Shift** key while dragging a Positioned Box to move it horizontally or vertically.

Dynamically changing the position

A Positioned Box has the following attributes:

- **anchor** defines the page number (starting by 0) the box is placed on
- **offset-x** defines the horizontal position of the box relative to its container

- **offset-y** defines the vertical position of the box relative to its container.

These attributes can be set in a script. The following script dynamically changes the position of a Positioned Box in a Print context by setting the offset-x and offset-y values.

```
results.attr('offset-x','96');
results.attr('offset-y','96');
```

The measurements are in pixels (e.g. 96px = 1in). Note that you do not need to set the units.

> **Note**
>
> Do not set the `top` or `left` property of a Positioned Box in a style sheet. The position of a Positioned Box in a Print context is handled via its attributes to take the page (or Master Page) and page margins into account. Attributes cannot be overwritten from within a style sheet: style sheets specify style properties, not values of attributes.

Styling a positioned box

A Positioned Box can be styled using the **Format > Box** menu item, through the CTRL+M keyboard shortcut, or through CSS; see "Styling and formatting" on page 294 and "Styling templates with CSS files" on page 296.

Inline Box

An Inline Box is one that is placed within the text flow, where other elements (including text) can wrap around it. An inline box is actually a element that is **floating**; in other words, it has its CSS property `float` set to `left`, `right` or `no float`.

Inline Boxes can be used in Print context and in Web pages. It is common to do entire web layouts using the `float` property. In Email templates, it is best to use Tables to position elements.

Adding an Inline Box

To insert an inline box, use the icon on the toolbar. Inline Boxes can be resized using the handles on the sides and corner. They can be styled using the **Format > Box** menu item, through the CTRL+M keyboard shortcut or through the CSS files; see "Styling and formatting" on page 294 and "Styling templates with CSS files" on page 296.

Positioning an Inline Box

Initially an Inline Box will float to the left. Use the (Float left), (No float) and (Float right) icons on the toolbar to change the position of an Inline Box within the text.

- The **Float left**button aligns the Inline Box to the left. The text is positioned to the right of it and is wrapped around the box.
- The **Float right**button aligns the Inline Box to the right, with the text wrapped around it to the left.
- The **No float** button positions the Inline Box where it occurs in the text.

It is not possible to move an Inline Box using drag and drop. To move the Inline Box to another position in the text, you have to edit the HTML on the Source tab in the Workspace, moving the element using cut and paste. To open the Source tab, click it (at the bottom of the Workspace) or select **View > Source View**.

Span

The Span element (in HTML code) is used to group inline elements, such as text in a paragraph. A Span doesn't provide any visual change by itself, but it provides a way to target its content in a script or in a style sheet.

To wrap content in a span, select the text and other inline elements and click **Insert > Wrap in Span** on the menu. Give the span an ID if you are going to add a style rule or script for it that is unique to this span; or give the span a class if this span can be targeted by a style or script along with other pieces of content. Now you can use the wrapper's ID or class as a script's or style's **selector**; see "Using the Text Script Wizard" on page 353 and "Styling and formatting" on page 294.

Div

The Div is the element used to create both Positioned Boxes and Inline Boxes. By default, a Div element reacts pretty much like a paragraph (<p>) or an inline box set to 'no float' except that it can be resized directly. Just like Positioned Boxes and Inline Boxes, Div elements can be styled using the **Format > Box** menu item, through the CTRL+M keyboard shortcut or through the CSS files; see "Styling and formatting" on page 294 and "Styling templates with CSS files" on page 296.

Adding a Div element

To add a Div, select **Insert > Structural Elements > Div** on the menu. For an explanation of the options, see "Inserting an element" on page 194.

HTML tag: div, span

When you add elements, such as text, images or a table, to the content of a template, you are actually constructing an HTML file. It is possible to edit the source of the HTML file directly in the Designer; see "Editing HTML" on page 193.

In HTML, boxes are elements. Spans are elements. To learn how to change the attributes of elements, see "Attributes" on page 193.

Business graphics

Business graphics are a great way to visualize data in your documents. Connect Designer lets you create three types of grapics: Bar Charts, Line Charts and Pie Charts. The charts are generated by the integrated **amCharts** library (https://www.amcharts.com/javascript-charts/).

This topic explains how to create and edit a business graphic using a **Wizard**, without scripting. The wizard helps you to select data and it shows a preview of both the chart and the supplied data.
Connect supports the majority of features of the amCharts library (version: 3.21.12), whether or not they are directly available in the Chart wizard or via the Chart Properties dialog. As a rule, configuration options that aren't available as an option in a dialog can be set manually on the Source tab of the Chart Properties dialog.

Instead of using a Wizard, you could insert a business graphic **manually** by turning a standard element into a chart object using scripting.
There is a How-to that guides you through that process and gives a better understanding of the underlying configuration: Programmatically configure a Chart object.
For another example see this How-to: Put one slice per detail record in a Pie Chart.

> **Note**
>
> As of PrintShop Mail Connectversion 2018.1, the way charts can be compiled and presented has been greatly improved. As a consequence, charts made with a version of Connect prior to 2018.1 may not be converted correctly when opened in a later version.

For a list of known backward compatibility issues see: "Business Graphics: Backward Compatibility Issues introduced in 2018.1" on page 96. Also note that charts based on expanded, custom scripts are never converted.

Inserting a business graphic

The Connect Designer has wizards for adding three kinds of graphics: Pie Charts, Bar Charts and Line Charts. These wizards insert the chart and its accompanying script, after you select the data that should be displayed, their colors and labels, and a few display options.

To insert a business graphic in a template:

1. Place the cursor where the graphic should be added. Also make sure that you have loaded data (see "Loading data" on page 341).
2. Click the toolbar button of the type of chart you want to add, or select **Insert > Business graphic** and choose the chart type.
   The wizard opens with a page on which you can specify the attributes and insertion point of the graph.
3. An **ID** is required. You can change the given ID and, optionally, add a class.
4. Check the option **Absolute** if you want the chart to have an absolute position. Leave it unchecked to insert the chart as part of the text, and use the **Location** drop-down to select where in the text the graphic should be inserted:
   - **At cursor position** inserts it where the cursor is located in the template.
   - **Before element** inserts it before the HTML element in which the cursor is currently located. For example if the cursor is within a paragraph, the insertion point will be before the tag.*
   - **After start tag** inserts it within the current HTML element, at the beginning, just after the start tag.*
   - **Before end tag** inserts it within the current HTML element, at the end, just before the end tag.*
   - **After element** inserts it after the element in which the cursor is currently located. For example if the cursor is within a paragraph, the insertion point will be after the end tag of the paragraph ( ).*
   - **Replace** inserts it in place of the currently selected element. (This option is not available when inserting content in a Master Page.)

* If the current element is located inside another element, use the **Elements** drop-down to select which element is used for the insertion location. The list displays every element in the breadcrumbs, from the current selection point down to the root of the body.
Note: HTML has some restrictions as to which types of elements are allowed as children of other elements. For example, a paragraph element is not allowed to have children that are block level elements - like a Div or a Table. If inserting content at the selected location would produce invalid HTML the final result may be different than expected. For example, when you insert a Div into a paragraph, the paragraph gets split in two. This means you end up with two paragraphs with the Div in between.

5. Click **Next** and specify the data for the graph. These settings can also be edited when you open the script, after you've inserted the chart. For an explanation of the options in the wizard, see "Selecting data for a Business Graphic" on the next page.
6. For a Bar Chart or Line Chart, you can click **Next** and set some basic parameters of the chart:
   - **Stack Series**: Check to stack the values in one bar, instead of having one bar per value. This is only useful with a chart that is based on a data table.
   - **Rotate**: Check to flip the x and y axes, so that the bars are horizontal starting from the left.
   - **Show Legend**: Check to show the labels of the selected data in a legend.
     This is only useful with a chart that is based on a data table. In a chart based on data fields, the labels of the fields appear under the x-axis.
   - **Titles**: Type the title that should appear near the horizontal (x) axis and/or the vertical (y) axis. (Note that the axes with their titles are exchanged when you rotate the chart.)

   These are just of few of the numerous display options that can be edited via the Chart Properties dialog; see "Enhancing a charts' design" on page 264.
7. Click **Finish** to insert the chart into the template.

The wizard adds the following to the template:

- A Div element. It has a `data-amchart` attribute, as you can see when you select the chart and open the Source view in the Workspace. The `data-amchart` attribute contains settings for how the data is displayed. These settings are made via the chart's properties (see "Enhancing a charts' design" on page 264).
  - A script. The script determines which data are displayed in the chart, with which colors and labels. The script can be edited any time; see "Selecting data for a Business Graphic" on the next page.
    When a preview or output must be generated, the script adds the data for the chart

to the `data-amchart` attribute (in a `dataProvider` property). Connect then passes the value of that attribute to the integrated amCharts library.

Rasterizing a business graphic

Business graphics are output as SVG images, but not all clients may support that format. Before generating output, you may want to 'rasterize' it. This converts the business graph into a JPG or PNG image.
To rasterize a business graphic, right-click it and select **Rasterize Options**.
For a JPG image you can set the quality of the resulting image in a percentage.

> **Note**
>
> A business graphic in an Email section is rasterized by default and output as PNG image, because email clients usually don't support SVG images.
> SVG images in an Email section give an error in the Preflight window (see "Doing a Preflight" on page 405).

Selecting data for a Business Graphic

A Business Graphic script determines which data, colors and labels are used in the chart. Double-click the script on the **Scripts** pane to open it.

> **Tip**
>
> To find the script that provides data for a chart, hover over the names of the script in the Scripts pane. That particular script will highlight the chart in the template and its selector is the same as the ID of the business graphic (preceded by #).

1. Next to **Values**, select **data fields** with a numerical value.

   > **Tip**
   >
   > The **Toggle non-numeric fields** button filters non-numeric fields from the list. The list will then display only Integer, Float and Currency data fields.

2. Adjust the **label** of each of the selected data fields as needed: click on the label and type the new one.
   In Bar and Line Charts with Data Fields as input data, these labels appear under the x-axis of the chart.

   > **Note**
   >
   > Labels are used in the legend. They will be visible when the legend is enabled.

3. Select a **color** for each of the selected fields. Click on the color to open the Edit Color dialog (see "Color Picker" on page 492).

   > **Note**
   >
   > Colors defined in the Chart Properties dialog override the colors set in the script. To open the Chart Properties dialog, right-click the chart after adding it, and select **Chart...**. See: "Enhancing a charts' design" below.

4. Use the **Move Up** and **Move Down** buttons to change the order of the fields, which is reflected in the chart.

The Preview shows what the chart looks like with the current settings, whilst the Data Preview shows which data will be passed to the amCharts library.

Enhancing a charts' design

Charts inserted with a wizard always have the same layout. The **amCharts** library that is integrated in Connect has loads of options to layout charts differently. Here's how to make use of those options.

Start by opening the Chart Properties dialog. Right-click the chart (in the template, or in the Outline pane) and select **Chart**.

Every tab menu in the Chart Properties dialog, except the last one, gives direct access to a number of layout options. For a description of the options see:

- "Bar Chart Properties dialog" on page 479
- "Line Chart Properties dialog" on page 507
- "Pie Chart Properties dialog" on page 523

Adding and editing properties manually

The last tab menu in the Chart Properties dialog, the **Source** tab, reflects the choices made in the other tabs. More importantly, this tab gives you the possibility to add any amCharts configuration option that is unavailable via the other tab menus.

On the Source tab, all settings are given in JSON. For example:

```
{
"type": "pie",
"legend": {
        "enabled": false
},
"radius": "100",
"innerRadius": "30"
}
```

(The `innerRadius` option, found on the Pie tab in the Pie Chart properties, gives this Pie Chart the shape of a donut.)
Note that only properties that were modified via one of the tab menus are included in the JSON on the Source tab.

To change the chart, you can simply edit the JSON.
For example, adding `"handDrawn": "true"` (at the same level as the "type" property) will distort the lines of a Bar chart or Line chart, producing a hand-drawn effect.
All configuration options can be found in the amCharts **API** documentation: https://docs.amcharts.com/3/javascriptcharts/.

It depends on the class to which a property belongs, where in the JSON the property should be added. Chart properties should be added at the highest level; for example:

```
{
"type": "serial",
"rotate": true,
...
}
```

Properties of the Legend (listed here: https://docs.amcharts.com/3/javascriptcharts/AmLegend) should go in the `legend` section in the JSON:

```
...
"legend": {
```

```
 "position": "right"
},
```

The Source tab also lets you change properties that *are* available in either the Script Wizard or other tabs of the Chart Properties dialog. You could, for example, set the colors of the bars, lines or slices by adding an array of hexadecimal color values, like this:
`"colors": ["#FF6600", "#FCD202", "#B0DE09", "#0D8ECF"]`
Note that properties defined on the Source tab override those made in the Script Wizard or on other tabs of the Properties dialog.

For inspiration you could use amCharts' **online editor**: https://live.amcharts.com/new/.
Properties can be copied directly from the Code tab in the live editor to the Source tab of the Chart properties dialog in Connect.
Note that copying the entire content of the Code tab will also carry over the sample data from the live editor (the `dataProvider` key). These will be overwritten by the chart script in Connect.

Note

- Properties defined on the Source tab override those made in the Script Wizard or on other tabs of the Properties dialog. The only exception is the `dataProvider` property: this will be overwritten by the chart's script.
- In Connect, the implementation was tested with Pie Charts (amPieChart) and with Bar Charts and Line Charts (amSerialChart). Other variants may or may not work.
- Animations will not work in the output, even if the output is a web page. This is because the chart is generated on the server, not in the browser.
  To get animations to work you would have to implement a solution similar to the one described in this How-to: Dynamic dashboard charts.

Using themes

The amCharts library supports working with themes. The default themes are: light, dark, black, patterns, and chalk. All except the 'patterns' theme can be used in Connect templates. Here's how to do that.

1. Add the theme to the top of the JSON on the Source tab of the Chart Properties diaog. For example:
   ```
   { "theme": "light",
   ...
   ```
   This setting overrides any color settings made in the Chart Script wizard and on the other

tabs of the Chart Properties dialog.

2. The 'light' theme requires no other settings. For the other themes you will have to manually set the background color of the Div element that holds the chart:
   A. Switch to the Design mode.
   B. Right-click the chart area and select **Box...** from the contextual menu.
   C. On the **Background** tab, set the **Color** to:
      - #282828 for the 'dark' theme and the 'chalk' theme
      - #222222 for the 'black' theme
3. Finally, the 'chalk' theme requires adding a remote stylesheet with this URL: 'https://fonts.googleapis.com/css?family=Covered+By+Your+Grace' to your template. See "Using a remote style sheet" on page 299.

Date

The Date element inserts the current system date, optionally making it dynamic so that it updates whenever the template is viewed or produces output.

Adding a date

To add a Date element, use the **Insert > Date** option in the menus. A dialog appears with the following controls:

- **Language**: Use the drop-down to select which language the date should be displayed in.
- **Update Automatically**: Check to update the date automatically when the template is viewed or produces output. When this option is checked, a placeholder is inserted in the template and a script is created to update it automatically, otherwise a static text with the date is inserted.
- **Available Formats**: Select the date/time format in which to display the date.

Click OK to insert the date or Cancel to close the dialog.

> **Tip**
>
> If you are looking to add a date that originates from a record set, to a template, see: "Variable Data" on page 351. To insert a date you could use either the drag and drop method or the Text Script Wizard; the latter lets you set the date/time format.

Changing the date

Once inserted, a date can be modified directly in the template (if it does not update automatically) or through the **date** script (if it does update automatically). To modify the date in the script:

1. Double-click the **date** script in the **Scripts** pane.
2. Between the round brackets after Date, enter the desired date in the following order: year, month, day, and optionally hours, minutes, seconds, milliseconds (see https://www.w3schools.com/js/js_dates.asp and https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Date.) When the time is omitted, it defaults to 12:00:00 AM.

Formatting an automatically updating date

The script added to automatically update the date uses the short date format. To change this:

1. Double-click the **date** script in the **Scripts** pane.
2. Delete the first line of the script.
3. On the second line, delete what comes after `format` and change `format` to `formatter` (see "formatter" on page 817).
4. Now type a dot after `formatter`, press **Ctrl** + **space** and choose one of the functions to format a date and time; see "Date, date/time and time functions" on page 818.

> **Note**
>
> The Locale, set in the **Edit > Locale** dialog, has an influence on the formatting of a date. The Locale can be the system's locale, a specific locale, or it can depend on the value of a data field or runtime parameter; see "Locale" on page 336.

Hyperlink and mailto link

Links can be added to any template but they only work in electronic output (web pages, email and PDF files). They can be a regular hyperlink pointing to a web page or a mailto link that will open the default email client when clicked.

Note

Hyperlinks - both external and internal - are not preserved when printing multiple pages on the same sheet (see "Imposition options" on page 774).

HTML element: a

When you add elements, such as text, images or a table, to the content of a template, you are actually constructing an HTML file. It is possible to edit the source of the HTML file directly in the Designer; see "Editing HTML" on page 193.

The HTML tag of a hyperlink or mailto-link is <a>. This is sometimes called an anchor tag. For a list of attributes, see https://www.w3schools.com/tags/tag_a.asp.

Adding a hyperlink or mailto link

1. Select text or an image.

   Note

   Although it is possible, it is not advisable to add a Hyperlink to other elements, such as a Paragraph or Div. HTML 4 specifies that hyperlinks and mailto-links may only contain inline elements. Block elements, such as a Div, may not appear inside a link. HTML 5 states that the link "may be wrapped around entire paragraphs, lists, tables, and so forth, even entire sections, so long as there is no interactive content within (e.g. buttons or other links)"; see https://www.w3.org/TR/html5/text-level-semantics.html.

2. Click the **Insert hyperlink** button on the toolbar, or on the menu, select **Format > Hyperlink > Insert**.
3. Select **URL** to create an **external** hyperlink pointing to a web page or an **internal** hyperlink pointing to another page in the document; or select **Email** to create a **mailto**-link that will open the default email client when clicked.

4. For a **URL**:
    - **URL**:
        - **External** hyperlink: Enter a valid, well-formed URL to link to. It must start with the protocol, such as http:// or https://.
        - **Internal** hyperlink: Enter an ID, preceded by a hash; for example: `#myID`. The link will point to an element with that ID.

        > **Note**
        >
        > An internal hyperlink may point to an element in another section. Just make sure that there is only one element in the document with the specified ID.

    - **Target**: use the drop-down or type in the target for the link.When the target is `_blank` the link will open in a new browser window or tab.

    For a **mailto link**:

    - **Email**: enter a valid email address that appears by default in the To: field of the email client.
    - **Subject**: type a default subject that appears in the Subject: field of the email client.
    - **Message**: type a message that appears by default in the Message field of the email client.

    Note that all these can be changed within the email client once the link is clicked.

> **Tip**
>
> To quickly change the text of a hyperlink, position the cursor on the link and click `~contents` in the Breadcrumbs. Now you can start typing the new text.

Dynamically inserting or modifying a hyperlink

You may wish to adjust a hyperlink depending on a value in a record that is merged to the template when generating output, for example, to provide a different mailto link for different customers.

Dynamically adding or modifying a hyperlink implies writing a script. For information about scripts, see "Writing your own scripts" on page 393.

Adding a personalized link

Personalized URLs (pURLs) are links that are tailor-made for a specific purpose, generally for individual clients. Typically, a pURL in a Connect template takes the user to a personalized landing page, for example, to download an invoice or get access to specific products or services. For more information, see Personalized URL.

Images

Images are a powerful ingredient in all of your templates. This topic explains how to add and use them. Currently the supported image formats are:

- BMP
- EPS
- GIF
- JPG/JPEG
- PDF
- PNG
- SVG
- TIF/TIFF

Base64 encoded images are supported; however, they are added differently: set the `src` attribute of an <img> element to the base64 code (see below, "HTML tag: img" on page 273).

For Email and Web output, **PNG** is the preferred image format.
EPS, PDF, SVG and TIFF images in an Email or Web section are automatically converted to PNG to ensure that they can be seen in the browser or email client.

32-bit BMP files are not supported.

Ways to use images

In templates, both **imported** images and **external** images can be used (see "Adding images" on page 273 and "Resources" on page 115). Once added to the content of a template, an

image can be resized (see "Resizing an image" on page 321) and alternate text can be linked to it (see "Setting an alternate text" on page 277).

Tip

Using images in an Email template? See "Using images in email campaigns: tips" on page 170.

Dynamic images

Images can be switched dynamically, so that a letter or email can include one image or another, depending on a value in the data set. Read "Dynamic images" on page 366 to find out how to add such switching images.

Background images

Several parts of templates, such as sections and media, and elements such as positioned boxes, can have a background image. Right-click the element and click the **Background** tab to select an image to be used as the element's background image. See "Background color and/or image" on page 323 and "Using a PDF file or other image as background" on page 142.

Tip

Editing PDF files in the Designer is not possible, but when they're used as a section's background, you can add text and other elements, such as a barcode, to them.
The quickest way to create a Print template based on a PDF file is to right-click the PDF file in the Windows Explorer and select **Enhance with Connect**. Alternatively, start creating a new Print template with a Wizard, using the PDF-based Print template (see "Creating a Print template with a Wizard" on page 126).
To use a PDF file as background image for an existing section, see "Using a PDF file or other image as background" on page 142.

Filling optional whitespace

Conditional content and dynamic tables, when used in a Print section, may or may not leave an empty space at the bottom of the last page. To fill that space, if there is any, an image or advert can be used as a 'whitespace element'; see "Whitespace elements: using optional space at the end of the last page" on page 149.

HTML tag: img

When you add elements, such as text, images or a table, to the content of a template, you are actually constructing an HTML file. It is possible to edit the source of the HTML file directly in the Designer; see "Editing HTML" on page 193.

In the section's source file, images are <img> elements. The <img> tag has at least four attributes: `src`, `alt`, `width` and `height`. `src` specifies the URL of the image. `alt` contains the alternate text; see "Setting an alternate text" on page 277.

The value of the attributes can be changed via a script; see "Attributes" on page 193.

Adding images

This section explains the difference between imported and external images and describes a number of ways to add images to a template.

Note

The Connect image cache size is limited to 100MB. This allows most output jobs to run faster. However, if a job requires more than 100MBs of image files, then the Connect image cache size can be increased to cater for such. Please contact OL Support for instructions on how to modify the image memory cache value, if needed. (See also: "Limit of 100MB of image files within a single job" on page 97.)

Note

An image with an unknown file extension is represented by a red cross in the output, but no error is logged unless the image refers to a local file that does not exist on disk. Image file extensions that the software recognizes are: AFP, BMP, EPS, GIF, JPG/JPEG, PCL, PDF, PNG, PS, SVG, and TIF/TIFF.

Imported or external images

In templates, both **imported** images and **external** images can be used.

Imported images are images that are saved within the template file. To import images into a template and add them to the content, you can use the drag-and-drop method or the Select Image dialog (both are explained below).

External images are either located on a specific website (URL), or in a folder on a hard drive that is accessible from your computer. Note that external images need to be available at the time the template is merged with a record set to generate output, and that their location should be accessible from the machine on which the template's output is produced.
External images are updated (retrieved) when the output is generated. To refresh them at any other time, use the Refresh option in the menu (**View > Refresh**) or the Refresh button at the top of the Workspace.
External images can not be added via the drag-and-drop method. Use the Select Image dialog instead (see below).
If you add an external image via the Source tab or via a script, and the URL doesn't have a file extension, you have to add the `filetype` parameter to the URL manually. Specify the file extension as its value, for example: `?filetype=pdf`, or `&filetype=pdf` if it isn't the first parameter. Note that the ampersand character needs to be encoded as `&`.

For information about referring to images in HTML or in a script, see "Resources" on page 115.

Via drag-and-drop

The drag-and-drop method is a quick way to import one or more images into a template.

1. Look up the image file or image files on your computer using the Windows Explorer.
2. Select the image (or images, using Shift+click or Ctrl+click) and drag the image file from the Explorer to the **Images** folder on the **Resources** pane at the top left.
3. To place an image in the content, drag it from the **Images** folder on the **Resources** pane to the content and drop it. The image will be inserted in the template at the position of the cursor.

Via the Select Image dialog

To either import an image into a template or use an external image in a template, the Select Image dialog can be used:

1. Position the cursor in the content where you want the image to be inserted.
2. On the **Insert** menu, click **Image**. Or, click the **Insert Image** button on the toolbar. The Select Image dialog appears.

- Click **Resources**, **Disk** or **Url**, depending on where the image is located.
  - **Resources** lists the images that are present in the **Images** folder on the **Resources** pane.
  - **Disk** lists image files that reside in a folder on a hard drive that is accessible from your computer. Click the **Browse** button to select a folder (or an image in a folder).
    As an alternative it is possible to enter the path manually. You can give a local path (e.g. C:\Images\Test.jpg) or use the "file" protocol. The complete syntax of a fully qualified URL with the "file" protocol is: `file://<host>/<path>`.
    Note: if the host is "localhost", it can be omitted, resulting in `file:///<path>`, for example: `file:///c:/resources/images/image.jpg`.
  - **Url** lists image files from a specific web address. Select the protocol (**http** or **https**), and then enter a web address (for example, http://www.mysite.com/images/image.jpg).

  Note

  If a URL doesn't have a file extension, and the option Save with template is **not** selected, the Select Image dialog automatically adds the `filetype` parameter with the file extension as its value (for example: `?filetype=pdf` (if it is the first parameter) or `&filetype=pdf`).
  The `filetype`, `page` and `nopreview` parameters are not sent to the host; they are used internally. Therefore, URLs that rely on one of these parameters cannot be used.

- With an external image, you can check the option **Save with template**. If this option is checked, the file will be inserted in the **Images** folder on the **Resources** pane at the top left.
  If it isn't saved with the template, the image remains external. Note that external images need to be available when the template is merged with a record set to generate output, and that their location should be accessible from the machine on which the template's output is produced. External images are updated (retrieved) at the time the output is generated.

3. If the image is a PDF file that consists of more than one page, select the desired page.

> **Note**
>
> The number of pages in a PDF file cannot be determined via the HTTP and HTTPS protocols. If you wish to use a page other than page 1 in a remote PDF, check the option **Save with template**; then click OK and reopen the dialog. Next, on the Resources tab, select the PDF, and select a page.

4. Click **Finish**. The image will be inserted at the position of the cursor.

Importing images from another template

To import images directly from another template, click **File > Import Resources...** in the menu. See: "Import Resources dialog" on page 502.

Using one file that contains a collection of images (a 'sprite')

When a template that contains lots of images is merged with a large record set, the many file requests may slow down the process of output generation. One solution is to combine the images into a single image file (an 'image sprite') and display the part that holds the image. This reduces the number of file requests and can improve the output speed significantly.

For an explanation of how to do this, see "Creating and using image sprites" on page 1028.

Base64 encoded images

To insert a base64 encoded image into a template, you need to go to the Source view, insert the <img> tag manually and set its URL - the `src` attribute - to the base64 string, for example: `<img src=""..."">`.
If the base64 string is present in one of your data fields, you can set the `src` attribute dynamically. For instructions see How to use the captured or selected image in a template.

Note that the image will not be visible in Design mode, but it will appear in Preview mode.

Moving an image

An image that is added to a section behaves like a character and is part of the text flow. To move it, simply click the image and drag and drop it somewhere else in the text flow. To learn

how to wrap text around it, see "Wrapping text around an image" on page 322.
How to make an image stay at a certain position (like any image added to a Master Page) is explained here: "Pulling an image out of the text flow" on page 322. When an image has an 'absolute position' it can be moved around freely: hover over the border of the image to get a move pointer, click that pointer and drag and drop the image somewhere else.

Styling an image

Images can be resized using the handles on the sides and corners, or via the Image Formatting dialog, which opens when you right-click the image and select **Image...**, or select the **Format > Image** menu item.
Images can be styled using the same dialog, or through the CSS files; see "Styling templates with CSS files" on page 296.

A number of issues related to image styling are discussed in a separate topic: "Styling an image" on page 320.

Just like many other elements, images can be given borders and rounded corners, they can have inner and outer margins and they can be rotated. How to do this is described in general formatting topics, such as "Border" on page 325 and "Spacing" on page 337. All general formatting topics are listed under "Styling and formatting" on page 294.

> **Note**
>
> It is recommended to resize images outside of the Designer, with image editing software.

Setting an alternate text

Once an image has been inserted in the content of a template, it can have an alternate text.

The alternate text will be shown in emails at the position of the image while the image is loading and when the image is not found.

To set an alternative text, click the image and enter the alternate text in the **Alternate text** field on the **Attributes** pane at the top right.

Using a CSS gradient to create an image

CSS gradients are a new type of image added in the CSS3 Image Module. CSS gradients let you display smooth transitions between two or more specified colors, while repeating gradients let you display patterns. This way, using image files for these effects can be avoided, thereby reducing download time and bandwidth usage. In addition, objects with gradients look better when zoomed in a browser, and you can adjust your layout with much more flexibility.

For more information about the various types of CSS gradients and how to use them, see https://developer.mozilla.org/docs/Web/CSS/CSS_Images/Using_CSS_gradients.

> **Note**
>
> When CSS repeating gradients are displayed in a PDF reader, artifacts, like very thin lines may occur. When this happens, try setting the gradient's position a little bit different.

Table

Tables serve two different purposes: they are a way to display data in a tabular format, and they are also a way to position elements on a page.
In HTML email, Tables are the most reliable way to position text and images; see "Designing an Email template" on page 167. In web pages, on the other hand, Inline Boxes are the preferred way to position elements. Tables should only be used to display data in a tabular format, not to position text and images. Tables used in web pages to position elements make those pages less accessible to users with disabilities and to viewers using smaller devices.
In print, Tables can be used for both purposes.

HTML element: table

When you add elements, such as text, images or a table, to the content of a template, you are actually constructing an HTML file. It is possible to edit the source of the HTML file directly in the Designer; see "Editing HTML" on page 193.

The HTML tag of a Table is <table>. Tables are divided into table rows with the <tr> tag. Table rows are divided into table data with the <td> tag. A table row can also be divided into table headings with the <th> tag.

The tags <thead>, <tbody> and <tfoot> can be used to group the header, body, or footer content in a table, respectively.

For information about HTML tables and a list of attributes, see https://www.w3schools.com/html/html_tables.asp.

Inserting a Table

1. On the toolbar, click the **Insert Table** button, or on the menu select **Insert > Table > Standard**.
2. Enter the Table's desired attributes:
   - **ID**: a unique identifier for the Table. IDs are used to access the Table from scripts and as CSS selectors for style rules.
   - **Class**: A class identifier for the Table. Classes can be shared between elements and are used to access the Table from scripts and as CSS selectors for style rules.
   - The number of **rows** for the header, body and footer of the Table.
   - The number of **columns**
   - The **width** of the Table.
3. Check the option **Absolute** to give the Table an absolute position, or use the **Location** drop-down to select where to insert the Table:
   - **At cursor position** inserts it where the cursor is located in the template.
   - **Before element** inserts it before the HTML element in which the cursor is currently located. For example if the cursor is within a paragraph, the insertion point will be before the tag.*
   - **After start tag** inserts it within the current HTML element, at the beginning, just after the start tag.*
   - **Before end tag** inserts it within the current HTML element, at the end, just before the end tag.*
   - **After element** inserts it after the element in which the cursor is currently located. For example if the cursor is within a paragraph, the insertion point will be after the end tag of the paragraph ( ).*
   - **Replace** inserts it in place of the currently selected element. (This option is not available when inserting content in a Master Page.)

   * If the current element is located inside another element, use the **Elements** drop-down to select which element is used for the insertion location. The list displays every element in the breadcrumbs, from the current selection point down to the root of the body.
   Note: HTML has some restrictions as to which types of elements are allowed as children of other elements. For example, a paragraph element is not allowed to have children that are block level elements - like a Div or a Table. If inserting content at the selected location

would produce invalid HTML the final result may be different than expected. For example, when you insert a Div into a paragraph, the paragraph gets split in two. This means you end up with two paragraphs with the Div in between.

Note

- Tables with an absolute position are only useful in Print sections.
- Tables on a Master Page have to have an absolute position, unless they are located inside another element with an absolute position.

4. Click **Next** and select the desired table style. The top left actually applies no styling to the table. The style can be easily changed afterwards; see "Styling a Table" on the next page.
5. Uncheck the box **Allow resizing** if the columns should not be resizable in the workspace (in Design mode). This is useful if the column size is determined in the Source mode or in a style sheet.
6. Click **Finish** to add the Table to the section.

Header and footer

Adding a header or footer

To add a header or footer to an existing Table, right-click the Table and then select **Table > Insert thead** or **Insert tfoot**, on the shortcut menu.

Alternatively, click in one of the cells and select **Insert > Table > Insert thead** or **Insert tfoot**, on the menu.

Deleting a header or footer

To delete a header or footer, simply right-click the header or footer and select **Row > Delete** on the shortcut menu.
If the deleted element was targeted by a script, you will be asked if you want to delete the script as well.

Rows and columns

Adding a row or column

To add a row or column to an existing Table, click in a cell. Then click the black triangle next to the **Insert Row Above** button on the toolbar, and click one of the **Insert** buttons, or select one of the options in the **Insert > Table Elements** menu.

Alternatively, right-click the Table and on the shortcut menu, select **Row > Insert Above** or **Insert Below**, or select **Column > Insert Before** or **Insert After**.

Deleting a row or column

To delete a row or column, simply right-click the row or column and select **Row > Delete** or **Column > Delete** on the shortcut menu. If the deleted row was targeted by a script, you will be asked if you want to delete the script as well.

Styling a Table

The Insert Table wizard lets you select a style for the table.

Tables are styled via CSS: when the wizard adds a Table, the chosen theme's **class** is added to the <table> element, and, if it doesn't exist yet, the **default_table_styles.css** file is added to the resources of the template. This CSS file contains the CSS rules for all table themes. (See: "Styling templates with CSS files" on page 296.)

To change the theme, you could simply select the table and change its class on the "Attributes pane" on page 577.
The available theme classes are: table--grid, table--colgrid, table--minimalist, table--dark, table--light, table--striped, table--topbar, table--portfolio. (Note the double dash in the class name.)

The default_table_styles.css file is read-only, but of course you could overwrite styles and create your own theme, using your own **style sheets**.
Regardless, you can change the font, font size and color, the borders, the cell padding (the distance between the edge of the cell and its content), and the background color or image of the table and its cells, via the **Formatting** dialog.

Both approaches are explained in the topic: "Styling a table" on page 316.

Resizing and moving a Table

Resizing a Table

- Select the Table (see "Selecting an element" on page 196) and type the desired width and height under **Geometry** on the **Attributes** pane.
- Select the Table and select **Format > Table**, on the menu. On the **Table** tab, change the **width** and **height** of the Table.
- Click in the Table and drag the handles to resize it. Press the **Shift** key while dragging to scale the Table proportionally.
  This option only works in a Print section, with a Table that has an absolute position and for which resizing is allowed.
  - If the position of the Table isn't absolute, right-click the Table and on the shortcut menu, select **Convert to absolute**. (This option isn't available for Tables on a Master Page, as they must always have an absolute position, or be located inside another element with an absolute position.)
  - Select the Table (see "Selecting an element" on page 196) and then, on the **Attributes** pane, check the option **Allow resizing**.

Moving a Table

The easiest way to move a Table in relation to other content is this:

1. Open the **Outline** pane (next to the Resources pane).
2. Select the Table on the Outline pane.
3. Drag and drop it somewhere else in the outline.

To move a Table with an absolute position, you can also:

- Click in the Table and then drag the border to move the Table.
- Select the Table (see "Selecting an element" on page 196) and type the desired values in the Top and Left fields on the **Attributes** pane.
- Select the Table and select **Format > Table**, on the menu. On the **Table** tab, change the **Positioning** values.

Hiding the border

When using a Table to position other elements, you will want to hide the borders of the Table. To do this, set the width of the border to 0; see "Border" on page 325.

Text and special characters

The vast majority of templates for personalized customer communications contain, of course, text. While the most common text element is a <p> or paragraph, other elements such as Headings (<h1> through <h6>) are also considered text elements. Text elements can be present within other types of elements such as table cells (<td>), boxes ( ), etc.

Adding text

To add text, simply type in the workspace in the middle.

- Press **Enter** to insert a new paragraph.
- Press **Shift+Enter** to insert a line break.

Alternatively, use the **Insert Lorem Ipsum** toolbar button to insert dummy text, or copy-paste text into the template.
Select **Paste as Plain Text** from the Edit menu or the contextual menu to insert the text without any HTML styles or formatting.

Text that precedes or follows the value of a **data field** can be added by the Text Script Wizard; see "Using the Text Script Wizard" on page 353.

Note: it is not possible to open a Word file in the Designer. When you copy text from a Word document, however, basic style characteristics travel with the content to PrintShop Mail Connect Designer. Formatting options like bold, italic and formats like Heading 1, Heading 2 are maintained.

Extra spaces

When you add spaces in Design or Preview mode the editor automatically preserves any extra spaces by converting them to non-breaking spaces (" " in HTML). It does this because in HTML extra spaces are generally removed. Take this into account when you edit the template in Source mode (i.e. in HTML) or add text via a script.

This how-to describes another way to maintain extra spaces in the text: Maintain extra spaces in text.

Adding special characters

To add special characters:

1. Position the cursor where the character should be inserted.
2. On the **Insert** menu, point to **Special Characters** click **Symbols**, **Dashes and Spaces**, **Arrows**, or **Geometric Shapes**, and click one of the available special characters.

Adding page numbers

Page numbers can only be used in the Print context. See "Page numbers " on page 150.

HTML element: p, h, li and others

When adding elements, such as text, images or a table, to the content of a template, you are actually constructing an HTML file, which you can see and edit on the Source tab (see also: "Editing HTML" on page 193).

In HTML, text can be contained in many different elements: paragraphs, span elements, line items and table cells, for example.

The HTML tag of a paragraph is . The paragraph should be followed by a closing tag: .

A line break looks like this in HTML: <br />.

Only the Source tab requires that you write HTML. In Design or Preview mode the editor adds HTML tags as needed.

Formatting text

Text can be styled, colored, centered, indented etc. It can even be displayed so that it reads from right to left. See "Styling text and paragraphs" on page 308.

In all templates you can use the fonts that are provided with the Designer, as well as imported fonts; see "Fonts" on page 332.

Translating text

OL Connect templates can be **multilingual**. For information about multilingual templates and how to create them, see "Translating templates" on page 467.

Snippets

A snippet is a small, ready-to-use piece of content in a file. Snippets can be re-used within the same template, in all contexts, sections and scripts.
There are three types of snippets:

- **HTML** snippets can contain any contents that a section can have, such as text, images, variable data, dynamic tables, etc., see "HTML snippets" on page 288.
- **JSON** files contain data; see "JSON snippets" on page 290
- **Handlebars** templates can contain HTML and Handlebars expressions; see "Handlebars templates" on page 291.

The information in this topic applies to all or at least two types of snippets.

> **Note**
>
> Regarding **styling**: when a snippet is added to different sections or contexts, it is displayed according to the section's or context's style sheet. This means that the same content can look different depending on the styles applied to the section or context, without changing the content.

> **Tip**
>
> It is possible to open and edit any external HTML or JSON file in the Designer: select **File > Open**, select **All files (*.*)** as the file type and then select an HTML or JSON file.

Adding a snippet to the Resources

Here's how to add a snippet to the Resources of a template.

1. Before adding a snippet, import any resource files that are related to the snippet, such as image files and CSS files, into the template file. Drag and drop the files to the corresponding folders (**Images** and **Stylesheets**, respectively) on the **Resources** pane. If you want to use external images, see "Images" on page 271.
2. Drag the file that contains the snippet into the **Snippets** folder on the **Resources** pane. If the added snippet is not UTF-8 encoded you will be asked to select its encoding. The snippet will then be converted and saved in the template as UTF-8 encoded file.

> **Tip**
>
> To **export** a snippet from your template, drag or copy/paste it out of the Snippets folder to a folder on the local hard drive.

Remote snippets

A remote snippet is either an HTML or JSON file that is not located within your template file but is hosted on a Content Management System or other external location.

To add a remote snippet:

1. Right-click the **Snippets** folder on the **Resources** pane, and click **New Snippet** > **Remote HTML file** or **Remote JSON file**.
2. Enter a name for the file as it appears in the Snippets folder. This name is shown in the Snippets folder with the **.rhtml** file extension for HTML, or **.rjson** for a JSON file.
3. Enter the URL for the remote resource. This must be a full URL, including the http:// or https:// prefix, domain name, path and file name.

> **Note**
>
> Remote snippets may contain other resources, such as images. There is one limitation though: only **absolute** paths are supported inside remote snippets. Images and other resources referred to with a relative path (for example: images/img.gif) or root-relative path (any path starting with a slash, for example: /base/images/img.gif) won't be available in the template.

Note

Remote snippets are expected to be UTF-8 encoded.

Creating a snippet

To start creating a snippet from scratch:

1. On the Resources pane, right-click the Snippets folder and select New.
2. Select the type of snippet that you want to create: **HTML** file, **JSON** file or **Handlebars template**.
3. Give the snippet a name.

Tip

It is also possible to create a new HTML snippet from an existing piece of content in the template; see "Using content to create an HTML snippet" on the facing page.

Adding a snippet to a section

Each type of snippets is used differently; see "Adding an HTML snippet to a section" on page 289, "Using a JSON snippet" on page 290 and "Using Handlebars templates: examples" on page 294.

Editing a snippet

To edit a snippet, open the snippet file by double-clicking it on the **Resources** pane.

By default, an **HTML snippet** that is being used as shared content can also be modified in a section where it is used. This behaviour can be changed; see "Editing a snippet" on page 289

Renaming a snippet

To rename a snippet, right-click it on the **Resources** pane in the **Snippets** folder and select **Rename**.

If you rename an **HTML snippet** that was inserted into a section as shared content, you need to update the reference to the snippet manually; see "Renaming a snippet" on page 290.

Translating a snippet

Snippets in a multilingual template get translated at the moment they are inserted in the output, if the text is tagged for translation. For more information see "Translating templates" on page 467 and "Tagging text in snippets" on page 471.

HTML snippets

HTML snippets can contain any contents that a section can have, such as text, images, variable data, dynamic tables, etc. Just like other types of snippets, HTML snippets are stored in the **Snippets** folder on the **Resources** pane, but their file name ends in **.html**.

This topic contains information that only applies to HTML snippets. If you want to know how to add an existing (remote) snippet to the resources, or how to create an empty HTML snippet, please see "Snippets" on page 285.

> **Tip**
>
> It is possible to open and edit any external HTML file in the Designer: select **File > Open**, select **All files (*.*)** as the file type and then select an HTML file.

Using content to create an HTML snippet

To turn a part of a letter or email into an **HTML** snippet for reuse in the content of a template:

1. Open the section and select the part or parts that should be saved to a snippet.
2. Right-click the selection, point to **Snippet** and click **Create** to **copy** the selected part to a new snippet, or **Create Shared Content** to create the new snippet and replace the selected part with a **reference** to the new snippet.
3. Give the snippet a name.

Adding an HTML snippet to a section

Drag-and-drop

To add an HTML snippet to the content of a section, drag the snippet from the **Snippets** folder on the **Resources** pane to the desired location in a section.
Check the option **Insert as shared content** to insert a **reference** to the original snippet in the template, rather than a **copy** of the original snippet.

Via a script

In addition to the drag-and-drop method, it is possible, and often useful, to insert a snippet or part of it, using a **script**; for remote snippets this is normal practice. See "Loading a snippet via a script" on page 420.

Editing a snippet

To edit a snippet, open the snippet file by double-clicking it on the **Resources** pane.

Editing shared content

By default, an HTML snippet that is being used as shared content can also be modified in a section where it is used. That way you actually adjust its source. The snippet will be changed at all locations where it is used.

The option to modify shared content snippets from within a section can be turned off:

- In the menu, select **View > Shared Content Editing** to enable or disable editing of all shared content.
- To enable or disable editing on a case by case basis, you can also manually set the `contenteditable` attribute on the Article element of the shared content to `true` or `false`. This overrides the menu setting.

> **Note**
>
> Remote snippets inserted as shared content cannot be changed from within the Designer. Of course, their source file can be edited outside of the Designer. When that happens, the changes will become visible in remote snippets that are inserted as shared content.

Renaming a snippet

To rename a snippet, right-click it on the **Resources** pane in the **Snippets** folder and select **Rename**.

If you rename a snippet that was inserted into a section as shared content, you need to update the reference to the snippet manually:

1. Open the section in which the snippet is used.
2. Switch to Source view.
3. Look for the <article> element in the HTML and replace the old snippet name with the new name in the source attribute.

Translating a snippet

Snippets in a multilingual template get translated at the moment they are inserted in the output, if the text is tagged for translation. For more information see "Translating templates" on page 467 and "Tagging text in snippets" on page 471.

JSON snippets

JSON snippets are snippets that contain **JSON data** instead of text. Just like other types of snippets, JSON snippets are stored in the **Snippets** folder on the **Resources** pane, but their file name ends in **.json**.

This topic contains information that only applies to JSON snippets. If you want to know how to add an existing (remote) snippet to the resources, or how to create an empty JSON snippet, please see "Snippets" on page 285.

> **Tip**
>
> It is possible to open and edit any external JSON file in the Designer: select **File > Open**, select **All files (*.*)** as the file type and then select an HTML or JSON file.

Using a JSON snippet

Unlike HTML snippets, JSON Snippets cannot be inserted into the content directly, but they can be accessed via a script using the `loadjson()` function.

To load a JSON snippet in script, use: `loadjson('snippets/nameofthesnippet.json')`.

For example:

```
var json_data = loadjson("snippets/snippet.json");
results.html(json_data.field1);
```

See also: "Writing your own scripts" on page 393 and "loadjson()" on page 827.

For an example in which JSON snippets are being used to localize a template, see this how-to: Localizing templates using json.

Handlebars templates

In OL Connect, Handlebars templates are a special kind of snippets. Just like other types of snippets (see "Snippets" on page 285), Handlebars templates are stored in the **Snippets** folder on the **Resources** pane, but their file name ends in **.hbs**.

Handlebars is the name of a JavaScript library that implements a templating language (see https://handlebarsjs.com/). It uses a template and an input object to generate output.

Handlebars templates look like regular text with embedded Handlebars expressions. A handlebars expression is a **{{**, some contents, followed by a **}}**. For example:
`<p>Hello {{firstname}} {{lastname}}!</p>`

When the template is rendered, these expressions are replaced with values from an input object.

The Handlebars library is **integrated** in OL Connect Designer. This means that you can use Handlebars expressions in this special type of snippets.

Working with Handlebars templates involves scripting. If you are new to scripting in the Designer, first read: "Writing your own scripts" on page 393.

> **Note**
>
> The information in this Online Help focuses on the implementation of Handlebars in OL Connect. For general information about Handlebars and how to use it, see the following

veb sites: https://handlebarsjs.com/ and https://devdocs.io/handlebars.

Creating a Handlebars template

To create a new, empty Handlebars template:

1. On the **Resources** pane, right-click the **Snippets** folder and select **New Snippet**.
2. Select the type of snippet that you want to create: **Handlebars template**.
3. Give the snippet a name.
4. Double-click the new file to open it in the Designer and fill it with HTML text and Handlebars expressions.

The editor for Handlebars snippets does not have a Design view. HTML with Handlebars expressions is not necessarily valid HTML. Processing it with an HTML parser might break both the Handlebars expressions and the HTML.

More about expressions and the functions that you can use in them can be found in the topic: "Handlebars expressions" on page 453.

Using a Handlebars template in a section

Although Handlebars templates contain HTML text, they cannot be inserted into the content of a section directly. The following steps need to be taken in order to replace Handlebars expressions with values and then add the content of a Handlebars template to a section.

Render the Handlebars template

First the template needs to be rendered, i.e. converted into HTML, replacing the expressions with values. This can be done with one line of code in a standard Designer script. Create a standard script and call the function:
`Handlebars.render(template, data)`.

For example:

```
var html = Handlebars.render( "snippets/policy-info.hbs", record );
```

The `template` can be:

- the name of a Handlebars template (.hbs) in the template
- the name of a Handlebars template (.hbs) on disk (file:///)
- the name of a remote Handlebars template (.hbs) (http:// or https://)
- a string that contains HTML and Handlebars expressions.

With a snippet on disk, the complete syntax is: file://<host>/<path>. If the host is "localhost", it can be omitted, resulting in **file:///<path>** - note the **three** forward slashes after **file:**.
In the remainder of the path you can either use escaped backward slashes:
"file:///C:\\Users\\Administrator\\Desktop\\Handlebars_LoadFile.hbs"
or forward slashes:
"file:///C:/Users/Administrator/Desktop/Handlebars_LoadFile.hbs"

The `data` can be the current record or part of it, or a JavaScript object.
If no data is passed, the **current record** will be used.

Add the HTML to a section

Finally, the rendered HTML can be added to the content of a section.
For example: `results.replaceWith( html )`

Alternative: compile and call the template

The `render()` function actually does two things:

1. **Compile the Handlebars template into a function**
   Handlebars templates need to be *compiled* first. When a Handlebars template is compiled, it is actually compiled into a JavaScript function.
2. **Call the function to replace expressions with data**
   The second step is to call the resulting function, passing in some data. The function will replace the expressions with values from the data and will then return HTML.

Instead of calling `Handlebars.render()` you could call `Handlebars.compile()` and then call the resulting function. For example:

```
var myFunction = Handlebars.compile( "snippets/policy-info.hbs",
record);
let html = myFunction( record )
```

Handlebars sample code from an online source will use the two separate steps since `Handlebars.render()` is only available in Connect.

See also: "Handlebars API" on page 464.

Using Handlebars templates: examples

Rendering a Handlebars template repeatedly

A Handlebars template is dynamic by nature. It is converted (compiled) to a JavaScript function. The Designer script that calls that function must pass data to it, and the result of the function will depend on that data. If the script calls that *same* function repeatedly, each time with different data - for example from a detail table -, the result of each call will be a different piece of HTML.

For example, this script loops over a detail table, each time passing different data to the same (compiled) Handlebars template.

```
let policies = record.tables.Policies;
let html = '';
for( var i = 0; i < policies.length; i++) {
{html += Handlebars.render( 'snippets/policy.hbs', policies[i] );
}
results.replaceWith( html );
```

Selecting a Handlebars template based on a data field

An approach that is also used with HTML snippets is to use the value of a data field in the current record to determine the name of the Handlebars template to use.

This is an example:

```
let html = ''
let policydata = record.tables.Policy
for (var i=0; i < policydata.length; i++) {
        let templateName = 'snippets/' + policydata[i].fields
['snippetName'] + '.hbs'
        html += Handlebars.render( templateName, policydata[i] )
}
results.replaceWith( html );
```

Styling and formatting

In the Designer you have everything at hand to make your templates look good: colors, fonts and all the tools to position, align and embellish elements in your designs. This topic informs

about the ways to style a template.

Local formatting versus style sheets

There are in general two ways to style elements:

- Using **local formatting**. Local formatting means styling an element directly, using a toolbar button or one of the formatting dialogs.
- Using **Cascading Style Sheets** (CSS). Style sheets can determine the appearance of individual elements, as well as the appearance of elements that have the same class or HTML tag.

Whether applied through style sheets or through local formatting, behind the scenes all layout properties in the Designer are CSS properties. When you format an element locally, an **inline** style rule is added to the element.

Note that where local formatting conflicts with a formatting rule for the same element in one of the style sheets, the local formatting rule gets priority; the rule in the style sheet will be ignored.

It is highly recommended to use style sheets in templates right from the start. Even more so if the communications are going to be output to different output channels, or if they consist of different sections (for example, a covering letter followed by a policy). Using CSS with templates allows a consistent look and feel to be applied. A style sheet can change the look of multiple elements, making it unnecessary to format each and every element in the template, time and again, when the company's layout preferences change. See "Styling templates with CSS files" on the facing page.

Layout properties

Colors and fonts make an important contribution to the look and feel of your template. See "Colors" on page 328 and "Fonts" on page 332.

Text and paragraphs have a number of formatting options that are not available for other elements: font styles and line height, for example. See "Styling text and paragraphs" on page 308.

Boxes and a number of other elements can have a background color and/or background image; see "Background color and/or image" on page 323.

Several elements, such as boxes, images, paragraphs, and tables, can have a border; see "Border" on page 325.

Boxes, images, tables, text and other elements can be rotated; see "Rotating elements" on page 315.

Spacing (padding and margin) helps to position elements relative to other elements in the template; see "Spacing" on page 337.

The best way to position elements depends on the output channel for which the template is intended; see "How to position elements" on page 312.

The locale setting influences how dates, numbers and amounts of money are displayed; see "Locale" on page 336.

Styling templates with CSS files

The Layout toolbar and the Format menu offer many possibilities to style every piece of a template. However, styling every single element, one after another, is a lot of work and, more importantly, can result in a template with a messy mix of styles that isn't easy to maintain and lacks consistent design. Therefore the preferred way to style templates is with CSS files: Cascading Style Sheets. This topic explains how to do that.

Why use CSS files

The basic idea behind CSS is to separate the structure and contents of a (HTML) document as much as possible from the presentation of that document.

Cascading Style Sheets were originally designed for use with web pages, or HTML files. Since every template in the Designer is constructed in HTML, CSS files can also be used in the Designer.
Instead of setting the font size, line height, color etc. for each and every paragraph in the template itself, you can define a layout for all paragraphs, and for all output channels, in a CSS file.

The benefit of this is that you can quickly and easily change the look and feel of all contexts in one template, without having to change the contents. In the event that your company chooses to use another font or to adjust its corporate colors, you only have to change the style sheets.

You are writing HTML

When you add elements, such as text, images or a table, to the content of a template, you are actually constructing an HTML file.

To see this, toggle to the **Design** tab in the workspace. Click anywhere in the content. Take a look at the *breadcrumbs* at the top of the workspace or select the Outline pane. The breadcrumbs show the HTML tag of the clicked element, as well as the HTML tags of other elements to which the clicked element belongs. The clicked element is at the end of the line.

To edit the HTML text directly:

- In the workspace, toggle to the **Source** tab.

On this tab you can view and edit the content of the template in the form of plain text with HTML tags (note the angle brackets: <>). You may add and edit the text and the HTML tags, classes, ID's and other attributes.

To learn more about HTML, see for example https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Introduction and http://www.w3schools.com/html/default.asp.

Many video courses and hands-on courses about HTML (and CSS) are offered on the Internet as well, some for free. Go, for example, to www.codeschool.com or www.codeacademy.com and look for HTML (and CSS) courses.

What you can't do with CSS

In Connect, it depends on the output channel what can and cannot be done with CSS. CSS can only be used to its full potential with HTML output. Animation and transition features won't work in Print output, obviously.

Included Cascading Style Sheets

When you create a template, a number of style sheets is automatically included:

- One style sheet that applies to all document types: context_all_styles.css.
- One or more style sheets specific to the context (Print, Email). For example, when you create an action email using the Wizard, the files context_htmlemail_styles.ccs and basic_email_action.css are automatically added to the **Stylesheets** folder on the **Resources** pane.

- A style sheet that defines default styles for tables: default.css. It contains the styles that you can choose from when you insert a table via the **Insert** menu or the **Insert table** toolbar button.

> **Note**
>
> Do not change the default.css style sheet. Use the global style sheet or the style sheet for the relevant context to define your own styles for tables.

Adding CSS files

To add a CSS file of your own, open an Explorer window, **drag** the file to the **Resources** pane and drop it on the **Stylesheets** folder.
In case the CSS file has references to specific images, you can drag/drop or copy/paste those images into the Stylesheets folder as well.

To create a new CSS file, right-click the **Stylesheet** folder on the **Resources** pane and select **New Stylesheet**.

You can also **import** one or more CSS file from another template using the "Import Resources dialog" on page 502.
Remember to add or import any files the CSS file refers to.

> **Note**
>
> The order in which style sheets are executed, can affect the actual output. This sequence can be set per section; see "Applying a style sheet to a section" on page 122.

> **Tip**
>
> - To export a CSS file from your template, drag or copy/paste it out of the Stylesheets folder to a folder on the local hard drive.
> - It is possible to open and edit any CSS file in the Designer: select **File > Open**, select **All files (*.*)** as the file type and then select a CSS file.

Using a remote style sheet

A remote style sheet is not located within your template but is located in a network folder or hosted on an external web server (generally called a **CDN**).

To add a remote style sheet:

1. Right-click the **Stylesheet** folder on the **Resources** pane, and click **New Remote Stylesheet**.
2. Enter a name for the file as it appears in the Stylesheet resources. For better management, it's best to use the same filename as the remote resource.
3. Enter the **URL** for the remote resource. If the file is located on an external web server, this must be a full URL, including the http:// or https:// prefix, domain name, path and file name. If the file is located on your network you can click the Browse button or enter the path and file name manually. The complete syntax with the "file" protocol is: `file://<host>/<path>`. If the host is "localhost", it can be omitted, resulting in `file:///<path>`, for example: file:///c:/resources/images/image.jpg.
   If the file is located on another server in your network, the path must contain five slashes after "file:".

> **Tip**
>
> After adding the remote file, you may right-click it and select **Download Resource**. This allows you to maintain a central file, from which you can quickly download a copy to your template without having to copy & paste.
>
> Note that a local copy of a remote resource is a snapshot; it is not automatically kept in sync with its remote content. You can download the remote resource again to overwrite the local copy with updated content. If you don't want a local copy to be overwritten you should rename it before downloading the remote resource again.

There are a few advantages to resources hosted on a CDN:

- These resources are not served by your server, saving on space, bandwidth and processing.
- Using a popular CDN takes advantage of caching - a client having visited another website using that same CDN will have the file in cache and not re-download it making for faster load times for the client.

Tip

To refresh the remote resources in a Designer view, use the Refresh option in the menu (**View > Refresh**) or the Refresh button at the top of the Workspace.

Using a Sass file

A CSS preprocessor is a CSS extension language that allows you to enhance CSS with code (variables, for example) and then compile it into plain CSS. CSS Preprocessor **Sass** is integrated in Connect.
For more information about Sass, see: https://sass-lang.com/.

For information about working with Sass in the Designer, see "Using a Sass file" on page 307.

Styling your templates with CSS files

Note

Email clients do not read CSS files and some even remove a <style> tag when it is present in the email's header. Nevertheless, CSS files can be used with the Email context in the Designer.
When generating output from the Email context, all CSS rules that apply to the content of the email can be processed and added either to the header of the email or to inline style properties as if local formatting was applied, depending on the Email section properties. See "Email section properties" on page 540.

Step 1: edit CSS

Editing CSS using a property sheet

1. Select **Edit > Stylesheets**.
2. Click the downward pointing arrow next to **Global** and select the context that you want to edit styles for, or select the Global CSS file to edit CSS rules that apply to all contexts.
3. Click **New**, or click one of the selectors that are already listed and click **Edit**.
4. Type a CSS selector. This can be:

- A class: `.class`. Class rules apply to all HTML elements with that class. When you create a class, choose a name that indicates what the class is used for, e.g. ‘small’ for a class that gives elements the font size ‘small’. The class name has to be preceded by a dot, e.g. `.small`.
- An ID: `#id`. An ID is always preceded by #, e.g. `#sender`. When you create an ID, choose a name that indicates what the ID is used for, e.g. #sender would refer to the HTML element with information about the sender.

> **Note**
>
> Each ID should be unique and can only be used once in each section.

> **Note**
>
> Do not give an element the ID 'pages' or the class name 'dynamic'. These are reserved words. Using them as an ID or class name leads to undesirable effects.

- An HTML element: p, h1, table, etc. Type the tag name without the angle brackets.
- A combination of HTML elements, separated by a comma. The CSS rule will apply to all HTML elements that are listed in the selector. For instance, a CSS rule with the selector `h1, p` applies to first level headings as well as paragraphs.
- HTML elements inside other HTML elements. For instance, a rule for all paragraphs inside a div element has the selector: `div p`.
- Etcetera. See https://www.w3schools.com/cssref/css_selectors.asp for more CSS selectors and combinations of CSS selectors.

5. Select the layout options that should apply to selected elements; see "Styling and formatting" on page 294. Note: where a width can be set as a percentage, it is a percentage of the space between the left and right margin.
6. Click the **Apply** button to see how a setting affects any elements that are subject to the selector. (You may have to move and resize the Stylesheet dialog before opening the Edit Rule dialog, in order to be able to see the template that you are working on.) If all is well, click **OK**.

7. In the Stylesheets dialog, click the selector that you chose. All CSS rules for that selector will become visible in a box below the list of selectors.

Editing plain CSS

- Click the button **Advanced** in any property sheet to open a CSS property editor. Type CSS properties at the left and values at the right.
- In the **Resources** pane at the left, double-click the global stylesheet or the stylesheet for the relevant context. The file opens in the workspace in the middle.

A list of all CSS properties and their possible values can be found here: https://www.w3schools.com/cssref/.
Custom CSS properties can be used as well; for an explanation and examples of these see: https://developer.mozilla.org/en-US/docs/Web/CSS/--*.

> **Tip**
>
> Right-click and choose **Format**, or press Ctrl + Shift + F to "pretty-print" the CSS and make it easier to read and edit.

> **Note**
>
> Block comments (/* ... */) are allowed in CSS, but single-line comments (// ...) are not standard and might not work as expected in all browsers.
> In the Designer, syntax highlighting doesn't work on single-line comments in CSS files.

Step 2: apply CSS to the content

After editing the CSS file(s), make sure that the CSS rules actually apply to one or more elements in the template.

- CSS rules for HTML elements, such as paragraphs, are automatically applied to all elements with the corresponding HTML tag.
- To make a CSS rule for a certain class or ID work for an element in your document, you have to add the class or ID to that HTML element (as described below).

> **Note**
>
> Classes may be reused throughout one section, but a specific ID should not be used more than once in each section. CSS layout rules for an element with a certain ID only apply to the first element with that ID in each section. If you have two sections inside of a Print context, then you can have the same ID on two sections; they will both be affected by the CSS rules for the element with that ID.

- Style sheets only apply to sections in which they are included; see "Applying a style sheet to a section" on the facing page.

Adding a class or ID to an HTML element

1. Select the element (see "Selecting an element" on page 196).
2. On the **Attributes** pane, type the **ID** and/or **class**. Type the ID **without** the preceding # and class names **without** a dot.

> **Note**
>
> Note: Elements can have multiple classes. Separate the class names with a space (eg. “red small”).

Alternatively, after selecting an element, you can click the **Source** tab at the bottom of the workspace. The selected element will be highlighted in the source. Add the class or classes and/or the ID to the opening tag of the HTML element, for example: `<p class=”intro”>`.

> **Tip**
>
> Would you like to know which elements have a certain class or ID?
>
> Use the Text Filter box at the top of the Resources pane. It allows to look for text in the source of any text-based files, including sections, master pages, and snippets.

Applying a style sheet to a section

In order for a style sheet to be applied to a specific section, it needs to be included in that section. There are two ways to do this.

Drag & drop a style sheet

1. Click and hold the mouse button on the style sheet on the **Resources** pane.
2. Move the mouse cursor within the **Resources** pane to the section to which the style sheet should be applied.
3. Release the mouse button.

Using the Includes dialog

1. On the Resources pane, right-click the section, then click **Includes**.
2. From the **File types** dropdown, select **Stylesheets**.
3. Choose which CSS files should be applied to this section. The available files are listed at the left. Use the arrow buttons to move the files that should be included to the list at the right.
4. You can also change the order in which the CSS files are read: click one of the included CSS files and use the **Up** and **Down** buttons. Note that moving a style sheet up in the list gives it **less** weight. In case of conflicting rules, style sheets read later will override previous ones.

Note

Style sheets are applied in the order in which they are included in a section. The styles in each following style sheet add up to the styles found in previously read style sheets. When style sheets have a conflicting rule for the same element, class or ID, the **last** style sheet 'wins' and overrides the rule found in the previous style sheet.

Note

Style sheets that are linked to (i.e. included in) a section show a chain icon in the Resources pane (see "Resources pane" on page 583).

How to determine which styles are applied

To see which styles are applied to an element, select the element (see "Selecting an element" on page 196) and take a look at the Styles pane that sits next to the Attributes pane.

> **Tip**
>
> Content added by a script isn't visible in Design mode, but is visible and can be inspected in Preview mode.

The Styles pane shows which CSS style rules apply to the currently selected element.
A link next to a style rule will open the file where that particular style is defined. This can be either a CSS file or the source file of a section if local formatting was used (see "Styling and formatting" on page 294).

A crossed-out style rule signals that it was overruled by another style rule. This happens when:

- A more specific, and therefore more important rule, is encountered for the same element. See "Using a more specific CSS rule" below to learn more about the specificity of style rules.
- A rule with the same importance is read after the first rule. Not only is the order of the rules in a CSS file important, but also the order in which the style sheets are read. The style sheets that are included with a section are read in the specified order; see "Applying a style sheet to a section" on page 122.

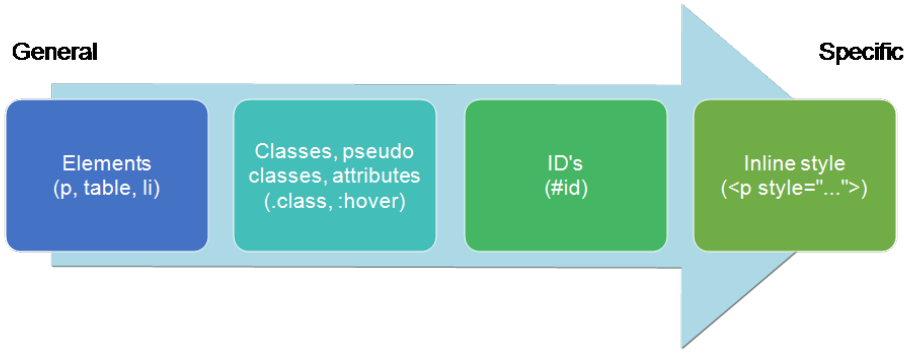
Using a more specific CSS rule

By default, many CSS properties of an HTML element also apply to the elements inside that element. For example, a CSS rule that specifies a certain font-type for a box also applies to paragraphs in that box. In this example the box is the 'parent' element and the paragraphs are the 'child' elements that inherit the font-type property of the box.

> **Note**
>
> Although the background color property seems to be inherited, it isn't. Most elements are transparent; therefore the background color of the parent element shines through.

To replace inherited style properties, you need to add a more specific CSS rule for that (type of) element. In case of a conflict between a general rule and a more specific rule, the more specific rule will be applied.

The following diagram shows the order of specificity.



Rules for HTML elements (p, table, li etc.) are general rules. Rules for classes, pseudo classes, and elements with a certain attribute (.class, :hover, [target]) are more specific. Rules for elements with a certain ID are even more specific. The most specific are inline styles.

Example

Assuming that a table has the CSS property "color: red" (which colors text in the cells red), a more specific rule for cells in that table could be, for example:

- A rule for the text color of all table cells (td elements), for example: `td { color: green; }`.
- A rule for the text color of table cells with a certain class, for example `.green { color: green; }`.
- A rule for the text color of a table cell with a certain ID, for example: `#greentext { color: green; }`.
- An inline style rule (local formatting) added to the HTML tag of a particular table cell, for example: `<td style="color: green;">...</td>`.

Each of these rules is more specific than the previous rules. All of these rules are more specific than the rule that applies to the table as a whole.

Note

When `!important` is added to a style rule (e.g. `color: red !important;`), this rule overrides any other style rules, even inline style rules.

Using a Sass file

A CSS preprocessor is a CSS extension language that allows you to enhance CSS with code (variables, for example) and then compile it into plain CSS. CSS Preprocessor **Sass** is integrated in Connect.
For more information about Sass, see: https://sass-lang.com/.

Adding a Sass file

To add a Sass file:

1. Right-click the **Stylesheet** folder on the **Resources** pane, and click **New Stylesheet > SCSS file**.
2. Enter a name for the file as it appears in the Stylesheet resources.
   When the name of Sass file begins with an underscore, it is considered a partial .scss file (e.g. _mySass.scss). Partial files are typically imported in a base .scss file. They may include Sass variables or other directives declared in the base file, and they cannot be compiled.

Compiling a Sass file

A Sass file needs to be compiled into a plain CSS file before it can be applied to a section. To compile it:

1. Open the **Stylesheet** folder on the **Resources** pane.
2. Right-click the .scss file and select **Compile**.

The compiled style sheet will have the same name as the Sass file, but with the extension .css. Compiled CSS files can be recognized by their first line: /* Compiled by https://sass-lang.com/libsass */.
Compiler options can be set in the Preferences; see "Editing preferences" on page 377.

Note

- Re-compiling a .scss file overwrites any manual changes made to the .css file.
- Partial .scss files cannot be compiled.
- Single line comments (//...) are not added to the compiled .css file, whereas multi-line comments (/* ... */) are maintained. Multi-line comments should be added within the brackets of the definition they apply to, to make sure they appear in the correct place in the compiled css.

Styling text and paragraphs

There are numerous ways to format text in a template. You can apply a certain font, make text bold, transform it to uppercase, center it, color it, etc.

This topic explains how to apply local formatting to text. It is recommended though, to format text using style sheets; see "Styling and formatting" on page 294 and "Styling templates with CSS files" on page 296.

Tip

With the **Copy fit** feature, text can automatically be scaled to the available space in a Box or Div. See "Copy Fit" on page 311.

Formatting text and paragraphs locally

An intuitive way of formatting text locally is by using the toolbar buttons: select some text, or an element that contains text (see: "Selecting an element" on page 196) and click one of the toolbar buttons to make it bold, center it, create a numbered or bulleted list, etc.

Tip

To quickly change a paragraph into a Heading, place the cursor inside of it, or select the paragraph (see: "Selecting an element" on page 196). Then select the appropriate element, either on the **Format** menu, or from the 'Element type' drop-down on the toolbar.

More local formatting options are available in the Formatting dialogs; see below.

Formatting text

To open the Text Formatting dialog, select some text and then select **Format > Text**. In the Text Formatting dialog you can set:

- The font, font size, color and background color:
  - **Font**: Select the font used to display text. See also: "Fonts" on page 332. This is equivalent to setting the `font-family` property in CSS.
  - **Font size**. Enter the size in a measure, named size or percentage. This is equivalent to setting the `font-size` property in CSS.
  - **Color**: Specify the color of the text: select a named color (defined in the "Colors Properties" on page 492) from the drop-down, or click the colored square to open the Color Picker dialog ("Color Picker" on page 492). Alternatively you could type a name or value in the Color field directly. It must be a predefined CSS color name (CSS color names), a hexadecimal color code (HTML Hex Color), an RGB color value, for example `rgb(216,255,170)` or a CMYK color value, for example `cmyk (15%, 0%, 33%, 0%)`.
    This setting is equivalent to the `color` property in CSS.
  - **Background color**: Specify the background color of the text: select a named color (defined in the "Colors Properties" on page 492) from the drop-down, or click the colored square to open the Color Picker dialog ("Color Picker" on page 492). Alternatively you could type a name or value in the Color field directly. It must be a predefined CSS color name (CSS color names), a hexadecimal color code (HTML Hex Color), an RGB color value, for example `rgb(216,255,170)` or a CMYK color value, for example `cmyk(15%, 0%, 33%, 0%)`.
    This setting is equivalent to the `background-color` property in CSS.
- The spacing between letters and words and the way the text is wrapped:
  - **Letter Spacing**: The space between characters in a text in measure or percentage. This is equivalent to the letter-spacing property in CSS.
  - **Word Spacing**: Set the space between each word in a text in measure or percentage. This is equivalent to the `word-spacing` property in CSS.
  - **Whitespace**: Specify how the text wraps. See CSS White-Space for details. This is equivalent to the `white-space` property in CSS.
- The style of the text. Check any option to apply the selected style to text within the element. This list shows the CSS property and value for each of the options:
  - **Bold**: Sets the `font-weight` to `700`.
  - **Italic**: Sets the `font-style` to `italic`.

- **Underline**: Sets the `text-decoration` to `underline`.
- **Strikethrough**: Sets the `text-decoration` to `line-through`.
- **Subscript**: Sets the `vertical-align` to `super`.
- **Superscript**: Sets the `vertical-align` to `sub`.
- **Capitalize**: Sets the `text-transform` to `capitalize`.
- **Uppercase**: Sets the`text-transform` to `uppercase`.
- **Lowercase**: Sets the`text-transform` to `lowercase`.
- **Small-caps**: Sets the `font-variant` to `small-caps`.

Note

All settings in the Text Formatting dialog are in fact CSS style rules. When you change one or more settings, the selected text gets wrapped in a Span element that has an **inline** style tag containing the selected setting(s). Click the **Advanced** button to add CSS properties and values to the inline style tag of the Span directly. For more information about CSS, see "Styling and formatting" on page 294.

Formatting a paragraph

Through the Paragraph Formatting dialog you can set the line height and first indent of a paragraph, and specify how to handle page breaks before, in and after the paragraph. It also lets you add spacing and a border; see "Spacing" on page 337 and "Border" on page 325.

To open the Paragraph Formatting dialog, select a paragraph (see: "Selecting an element" on page 196) or place the cursor in a paragraph, and then select **Format > Paragraph**.
For an explanation of all options in this dialog see: "Paragraph Formatting dialog" on page 520.

Removing local formatting from text

Layout buttons and options on the Format menu add **inline** style tags to the text. Style tags can look like this: **...** or like this: **<p style= "color: red;" >**.

Inline style tags have priority over styles defined in a CSS file because they are considered more *specific* (see "Using a more specific CSS rule" on page 305). For example, when a formatting rule in a style sheet colors all paragraphs green, a paragraph with an inline style tag to color it red would still stay red. So, when a rule in a style sheet doesn't seem to work, an inline style tag may be the culprit. In that case you might want to remove the local formatting.

To remove local formatting:

- Select the formatted text and click the toolbar button **Remove Formatting**. Doing this removes inline style tags from the selection.
- Alternatively, click the **Source** tab at the bottom of the workspace (or select **View > Source View**) to manually remove style tags.

> **Tip**
>
> When you select an element in the template, the **Styles** pane will show which styles are applied to that element. The link behind the style will take you to the place (the Source tab, or a CSS file) where that style is defined.

Copy Fit

Copy Fit is a feature to automatically scale text to the available space: the name of a person on a greeting card for example, or the name of a product on a shelf talker.
This feature is only available with Box and Div elements in Print sections.

Activating the Copy Fit feature

After adding a Box or Div element to a Print section (see "Boxes" on page 256), you can activate the Copy Fit feature on that element. Text inside that Box or Div or text in an element inside it, will then be scaled to fit the available space. This is how it's done:

1. Right-click the Box or Div element and click the respective element on the shortcut menu. Alternatively, select the element (see "Selecting an element" on page 196) and on the **Format** menu click the respective element.
2. Click the **Content** tab.
3. Check the **Copy Fit** option.
4. Enter the **Min font size and/or Max font size** using a valid font measurement unit (pt, px, in, cm or mm). Do not put a space between the number and the unit.
    - The **minimum font size** is 1pt. The default minimal font size is 4pt.
      When the minimum font size is left **blank**, the font size in Design view becomes the minimum font size. This means that the text can only be made bigger than its initial size.

- The **maximum font size** is 1048pt. By default this is 48pt.
  When the maximum font size is left **blank**, the font size in Design view becomes the maximum font size, so that the text can only be made smaller than its initial size.

5. When the option **Fit to width only** is checked, no line breaks will be added to the text.
6. Optionally, you can specify a **child** (an element inside the Box or Div) by giving its ID, for example: **#product**, or class, for example: **.product** - note the dot. The Copy Fit feature will only be applied to this child element.

> **Tip**
>
> To give an element a class or ID, select it and add the class or ID on the Attributes pane. An ID is meant to be used once in each section, while a class can be shared between several elements.

7. Click **Apply** or **OK**. Note that the effect of the Copy Fit feature can only be seen in Preview mode.

If it is impossible to make a text fit within the box with the given minimum and maximum font size, this will be reported as an error during a preflight (see "Testing scripts" on page 403).

How it works

When the Copy Fit feature is activated, the font size is calculated for the entire Box. Elements inside that Box get a font size relative to the Box. This means that their relative proportions are maintained.

How to position elements

To position elements in relation to each other in a template, you can wrap those elements in a Table or Box (see "Table" on page 278 and "Boxes" on page 256), and/or use the Spacing property of the elements. The Spacing property can also be used to indent elements or create a hanging paragraph or image; see "Spacing" on page 337.

Aligning objects with an absolute position is easy with the Alignment buttons. Guides help to align elements as well; see "Aligning objects" on the next page.

The top, right, bottom, and left properties specify offsets from the edges of the element's containing block.

Where to use Tables and Boxes

Tables, Positioned Boxes and Inline Boxes can help position elements in relation to other elements. It depends on the context which element is best to use.

In the **Email** context, Tables are the most reliable way to position text and images; see "Designing an Email template" on page 167 and "Table" on page 278.

In the **Print** context, Tables can be used to position elements, as well as both types of Boxes; see "Table" on page 278 and "Boxes" on page 256.

Spacing

Boxes, tables, paragraphs and many other elements have a **margin** and **padding**.
The margin is the white space around an element, outside the border. It is used to position an element in relation to the other elements, by putting more space between the element and its surrounding elements.
The padding is the space between an element's content and its border. It is used to position the content of the element inside the border.

To learn how to set an element's spacing properties, see "Spacing" on page 337.

> **Tip**
>
> Use a negative left margin to create a hanging paragraph or image.

Aligning objects

In Print sections, objects with an 'absolute position', such as a Positioned Box, have a fixed position in relation to the page (see also: "Using the CSS position property" on the facing page).
Objects with an absolute position can be aligned easily:

1. Press the **Ctrl** key and hold it down while clicking on the objects that you want to align. The **last** selected object is the reference object; this object will not be moved.
2. On the menu, select **Format > Align Objects**, or use the respective Toolbar buttons to align the objects.

Guides

Guides are horizontal and vertical lines used to help in designing templates, for example when positioning absolute positions boxes over a PDF background. They can only be used in Print sections.

- Select **View > Guides > Show guides** to show or hide the guides and margins.

To **add** a guide, press the **Insert Horizontal Guide** or **Insert Vertical Guide** buttons on the Toolbar.

To **move** a guide, click and drag it to a new location.
Click the **Shift** key while dragging to make the guide snap to the closest ruler tick.

Double-clicking a guide brings up its Edit dialog where its exact position can be adjusted.

- Select **View > Guides > Lock guides** to lock the guides in their current position.
- Select **View > Guides > Snap to guides** to make Positioned Boxes (and any other objects that have their **position** set to **absolute**) snap to guides when moved within a few pixels of them.

To **delete** a guide, double-click on it and press the **Delete** button.

Using the CSS `position` property

An element can be positioned independently of the text flow by changing its `position` property to `absolute` or to `relative` (that is, relative to the 'parent', its container).
When an element is placed inside another element, such as a Box, changing its `position` property to `absolute` positions the element absolutely inside its parent.
With the `position` property of an element set to `absolute`, the `top` or `bottom` and `left` or `right` properties position the element inside its parent with exact values: pixels (px), centimeters (cm), etc. Negative values are allowed.

> **Tip**
>
> You can quickly change the `position` property of an element in a Print section by right-clicking it and selecting **Convert to Absolute** or **Convert to Inline**.

For an explanation of all values that the `position` property can possibly have, see https://www.w3schools.com/css/css_positioning.asp.

Where to use it

In Print sections, setting the `position` property to `absolute` can be very useful. It takes the element out of the text flow, so that the element stays where it is on the page. On Master Pages (which are only used in Print sections) elements are always positioned absolutely; if not, they must be located inside an element that has an absolute position.

In Email sections, do not use this property. Use Tables instead (see "Designing an Email template" on page 167 and "Table" on page 278).

How to use it

In the Formatting dialog the `position` property can often be found on the first tab, under **Positioning**. To open the Formatting dialog, right-click the element and click the respective element on the shortcut menu. Alternatively, select the element (see "Selecting an element" on page 196) and on the **Format** menu click the respective element.

This property isn't present in one of the tab menus of the style rule editor, but you can add it and specify a value after clicking the Advanced button in the style rule editor (see "Styling templates with CSS files" on page 296).

About the CSS `display` property

The `display` property is one of the most important CSS properties for controlling layout. Yet it is unlikely that you will use it often to position elements in a template: in most cases, the initial value of the `display` property for an element will be the right one.
It is more likely that you will use this property in style sheets and scripts to hide certain elements, by setting the value of this property to `none`(`display: none;`). (See "Styling templates with CSS files" on page 296 and "Writing your own scripts" on page 393.)
For an online tutorial about this property, see w3schools website.

Rotating elements

In any type of template, boxes, images, tables, text and other elements can be rotated.

The toolbar buttons **Rotate Clockwise** and **Rotate Counter Clockwise** rotate the element in which the cursor is located 90 degrees at a time.

To rotate an element into another angle position, use the 'angle' CSS property of the element. In most cases, this can be done in the element's Formatting dialog. In other cases, such as with text, you have to enter the CSS property and value manually. Both methods are explained in the following procedure.

1. Right-click the element and click the respective element on the shortcut menu. Alternatively, select the element (see "Selecting an element" on page 196) and click the respective element on the **Format** menu.
2. On the first tab, look for the **angle** property. If it is available, type the number of degrees the element should be rotated. A positive number will rotate the element clockwise, a negative number rotates it counter-clockwise. Skip steps 3 to 6.
   If the angle property is not available, proceed with the following step.
3. Click the **Advanced** button to open the Advanced Formatting dialog.
4. Click in the first blank field under **Property** and type **transform**.
5. Click in the field next to it, under **Value** and type **rotate(**, followed by the number of degrees the element should rotate, and then **deg)**, for example: rotate(20deg). A positive number will rotate the element clockwise, a negative number rotates it counter-clockwise.
6. Close the Advanced Formatting dialog.
7. Close the Formatting dialog, or click the Apply button to see the effect without closing the dialog.

Note

It is also possible to rotate elements by creating a style rule in a style sheet; see "Styling templates with CSS files" on page 296.

Styling a table

Just as other elements, tables can be styled in two ways:

- With **local formatting**. This means styling the table directly, using the Formatting dialog.
- Via **Cascading Style Sheets** (CSS). In a style sheet, style rules are declared for elements with different HTML tags, ID's and classes.

These two methods are described below. See "Styling and formatting" on page 294 for background information about these two methods.

Selecting a table, row or cell

There are several ways to select a table or row:

- Click in the table or row. Then, in the **breadcrumbs** (see "Selecting an element" on page 196) click **table** to select the table, or **tr** to select the row.
- Right-click a cell and from the shortcut menu, choose **Table > Select** or **Row > Select**.
- Click in a cell and then use the toolbar: click the **Select Table** button or click the black triangle next to that button and then click **Select Table** or **Select Row**.

Selecting one cell is easy: just click in it.

> **Tip**
>
> Use the Styles pane to see which styles apply to the currently selected table, row or cell.

Styling a column

In HTML, column elements don't exist, only tables, rows and cells.
To make cells look like a column, make sure that they are positioned underneath each other - insert empty cells if needed -, and then style the cells as usual.

Via the Formatting dialog

The Formatting dialog allows you to change the font, font size and color (see "Fonts" on page 332), the borders (see "Border" on page 325), the margins or cell padding (the distance between the edge of the cell and its content, see "Spacing" on page 337), and the background color or image of the table and its cells ("Background color and/or image" on page 323).

To open the Formatting dialog for **one cell** or for the **table as a whole**:

- Click in a cell and choose **Format > Table** or **Format > Table Cell**.
- Right-click it and choose **Cell...** or **Table...** from the shortcut menu.

Note that in this case **Table** styles the table as a whole. When you choose **Table** and change the border, for example, the borders of the cells inside it will not be changed.

To style **all cells** in a table or row at the same time via the Formatting dialog, you have to select the table or row first; see "Selecting a table, row or cell" on the previous page. Next, to open the Formatting dialog, choose **Format > Table Cell**. The settings that you make now will be applied to all cells in the selected row or table.

For information about specific options in the formatting dialogs, see "Table Formatting dialog" on page 553 and "Table Cell Formatting dialog" on page 557.

Via a style sheet

Cascading Style Sheets (CSS) offer more ways to style a table and its contents, than the Formatting dialog does.

How to use style sheets is explained in another topic; see "Styling templates with CSS files" on page 296.

Note that to make a style rule apply to a **specific** table, row or cell, you have to add an ID or class to that table, row or cell.

A style sheet contains a bunch of style rules for different elements, that are identified via a CSS **selector**. This can be the element's HTML tag (without angle brackets), an attribute, its ID, or a class.

When used as a CSS selector, the HTML tag for a table is **table**. For a row, it is **tr** and for a cell, **td**. A style rule that uses one of these, however, would apply to **all** tables, rows, or cells. For a rule to be more specific you need to add an ID (for a unique element) or a class (for a set of similar elements) to the table, row or cell, and use that as the style rule's selector.

Adding an ID or class to a table, row or cell

Before you can add an ID or class to a table, row or cell, you have to select that table, row or cell (see "Selecting a table, row or cell" on the previous page). After selecting the cell, row or table, type the ID or class in the respective field on the **Attributes** pane.

In CSS, refer to the table, row or cell with `#ID` (where ID should be replaced with the actual ID) or with `.class` (where class should be replaced with the actual class).

Styling the first, last and nth rows

The CSS pseudo-classes `:first-child`, `:last-child` and `:nth-child()` are very useful for styling table rows.

A CSS **pseudo-class** follows a selector to specify a special state of that selector. It always starts with a colon.

The pseudo-classes `:first-child`, `:last-child` and `:nth-child()` select an element only if it is the first, last or nth child element respectively. (In HTML and CSS, the word **child** refers to an element inside another element.)

The following CSS style rule selects the table row (tr) that comes first (:first-child) in its parent (which naturally is a table), and colors its background red:

```
tr:first-child {
        background: red;
}
```

> **Tip**
>
> In a Dynamic Table, data are in the body of the table (selector: tbody) and subtotals are in the footer (selector: tfoot).

Selecting a specific row, odd or even rows, or every nth row

The pseudo-class `:nth-child()` lets you select a specific row, all odd or even rows, or every **n**th row.

Between the round brackets in `:nth-child()` you can fill in a number, `odd` or `even`, or a formula: `an+b`. In the formula, `a` represents a cycle size (every...), `n` is a counter (for the child elements), and `b` is an offset value ('start at b'). The following examples will make this clear.

`:nth-child(3)` matches just one element: the third child element.

`:nth-child(odd)` matches child elements 1, 3, 5, 7, etc. The keyword `odd` substitutes the expression `2n+1`, which in other words says: 'take every second element, starting at 1'.

`:nth-child(even)` matches child elements 2, 4, 6, 8, etc. The keyword `even` substitutes the expression `2n+0`, or simply `2n`.

`:nth-child(3n)` matches child elements 3, 6, 9, 12 etc.

`:nth-child(3n+1)` matches child elements 1, 4, 7, 10 etc., so every third element, starting at 1.

Via script (based on a data field value)

To style a table, row or cell based on a data field value, you have to write a script (see "Writing your own scripts" on page 393).

First add an ID or class to the table, row or cell that needs to be styled: select the element (see "Selecting a table, row or cell" on page 317) and add an ID on the **Attributes** pane. Then create a script, using that ID or class as the script's selector. The script can be very simple:

```
if (record.fields.COUNTRY == 'CANADA') {
        results.css('color','green');
}
```

The Designer Scripts API provides several functions to style elements, for example `css()`, `hasClass()` and `addClass()` (see "Standard Script API" on page 810).

For information regarding scripts for **Dynamic Tables**, see "Using scripts in Dynamic Tables" on page 426.

Styling an image

Just like many other elements, images can be given borders and rounded corners, and they can be rotated. How to do this isn't any different from the way it is done with other elements, so it isn't described in this topic, but in general formatting topics; see "Styling and formatting" on page 294.
This topic discusses specific image formatting issues.

Note that image characteristics like brightness and contrast can not be changed within the Designer.

Local formatting vs. style sheets

Just as other elements, images can be styled in two ways:

- With **local formatting**. This means styling the image directly, using the Formatting dialog.
- Via **Cascading Style Sheets** (CSS). In a style sheet, style rules are declared for elements with different HTML tags, ID's and classes.

See "Styling and formatting" on page 294 for background information about these two methods.

Applying local formatting to an image

To apply **local** formatting to an image, either:

- right-click the image and select **Image...** from the contextual menu
- click the image and select **Format > Image...** from the menu.

For an explanation of the available options, see "Image Formatting dialog" on page 500.

Applying style rules to an image

To format an image via a **style sheet**, first give the image an ID or class: select the image, and enter the ID or class on the Attributes pane. This makes it possible to make the CSS style rule target this image specifically, or a set of images with the same class. A style rule with the selector `img` (the HTML image tag) would apply to all images.
Next, create the style rule; see "Styling templates with CSS files" on page 296. Note that when a property isn't present in the style rule editor, it can still be used: click the Advanced button in the style rule editor; enter the property under Property, and its value under Value.

Resizing an image

There are several ways to resize an image after inserting it in the content of a template.

- Click the image and drag the handles to resize it. Press the **Shift** key while dragging, to scale the image proportionally.
- Select the image (see "Selecting an element" on page 196) and type the desired width and height in the respective fields on the **Attributes** pane.
- Select the image and select **Format > Image**, on the menu. On the **Image** tab, change the **width** and **height** of the image.
- Set the size of the image in a style sheet (see "Styling templates with CSS files" on page 296).

The size can be set in a measure or as a percentage of the containing element.

Reset the image size

To reset an image to its original size, select the image and click the **Reset Image Size** button on the Attributes pane, under Geometry.

Positioning an image

Wrapping text around an image

Initially, when an image is inserted into a paragraph, it behaves as if it were a character. Text isn't wrapped around an image automatically. To make that happen, you have to change the `float` property of the **image** to `left` or `right`. This anchors the image to the left or right, allowing text to be wrapped around it.

Select the image (see "Selecting an element" on page 196) and use the (Float left) and (Float right) icons on the toolbar to change the position of an image within the text.

- The **Float left** button aligns the image to the left. The text is positioned to the right of it and is wrapped around the box.
- The **Float right** button aligns the image to the right, with the text wrapped around it to the left.
- The **No float** button positions the image where it occurs in the text, as if it were a character. Text is not wrapped around it.

To position an image using the menu, select the image and then select one of the options in **Format > Float**.

Alternatively, open the Formatting dialog (see "Applying local formatting to an image" on the previous page): select the image; on the menu, select **Format > Image** and on the **Image** tab, under **Text Wrap**, set the **Float** property.

The `float` property could also be changed via a style sheet. This property isn't present in one of the tab menus of the style rule editor directly, but you can add it and specify its value after clicking the Advanced button in the style rule editor (see "Applying style rules to an image" on the previous page).

Pulling an image out of the text flow

When dragged into a template, an image is automatically integrated in the text flow. This means that it will move up or down, depending on the preceding text.
In a Print section, to position the image independently of the text flow, you can change its

`position` property to `absolute`. (For an explanation of all possible values for this property, see https://www.w3schools.com/css/css_positioning.asp.)
When an image is placed inside a Box (or Div element), changing its `position` property to `absolute` positions the image absolutely inside that Box.
Note that `float`, the property that can make an image float to the right or left (see "Wrapping text around an image" on the previous page), is a relative positioning property, since it specifies the position of the element relative to its container. This means it is incompatible with the `position:absolute` property.

In the Formatting dialog (see "Applying local formatting to an image" on page 321) the `position` property can be found on the **Image** tab, under **Positioning**.
The `position` property isn't present in one of the tab menus of the style rule editor directly, but you can add it after clicking the Advanced button in the style rule editor (see "Applying style rules to an image" on page 321).

When the `position` property of an element is set to `absolute`, the `top` or `bottom` and `left` or `right` properties can be used to position the element inside its parent with exact values (pixels (px), centimeters (cm), etc). Negative values are allowed.

> **Note**
>
> In Web sections, the `position` property may sometimes be useful for images inside a Div element, but generally elements should not be positioned absolutely. Designs for the Web should be flexible so that they display nicely on a variety of devices and screen sizes.

Background color and/or image

In any type of template, boxes, tables and table cells can have a background color and/or a background image.

To select a background image or color:

1. Right-click the box and click **Box** on the shortcut menu.
2. Alternatively, select the box (see "Selecting an element" on page 196; note that a Box is a element) and on the **Format** menu click **Box**.
3. Click the **Background** tab.

To define a **background color**:
Click the downward pointing arrow next to **Color** to select a color from the list of predefined colors (see "Background color and/or image" on the previous page).
Alternatively, click the small rectangle to the right of the color list to open the Color Picker dialog. In this dialog you can select a color from the color wheel. You can also choose the color mode: RGB or CMYK. For an explanation of these two modes, see "Background color and/or image" on the previous page; for an explanation of the other options in this dialog, see "Color Picker" on page 492.
You could also type a name or value in the Color field directly. It must be a valid color name (see color names on w3schools), a hexadecimal color code (see w3school's color picker), RGB color value, for example `rgb(216,255,170)` or CMYK color value, for example `cmyk(15%, 0%, 33%, 0%)`.

To select a **background image**:

1. Click the **Select Image** button.
2. Click **Resources**, **Disk** or **Url**, depending on where the image is located.
   - **Resources** lists the images that are present in the **Images** folder on the **Resources** pane.
   - **Disk** lists image files that reside in a folder on a hard drive that is accessible from your computer. Click the **Browse** button to select a folder (or an image in a folder). As an alternative it is possible to enter the path manually. The complete syntax is: file:///<host>/<path>. Note: if the host is "localhost", it can be omitted, resulting in file:///<path>, for example: file:///c:/resources/images/image.jpg.
   - **Url** lists image files from a specific web address. Select the protocol (**http** or **https**), and then enter a web address (for example, http://www.mysite.com/images/image.jpg).
3. With an external image, you can check the option **Save with template**. If this option is checked, the file will be inserted in the **Images** folder on the **Resources** pane.
   If not saved with the template, the image will remain external. Note that external images need to be available when the template is merged with a record set to generate output, and that their location should be accessible from the machine on which the template's output is produced. External images are updated (retrieved) at the time the output is generated.
4. Select an image from the list.
5. If the image is contained in a PDF file that consists of more than one page, select the desired page.
6. Click **OK**.

7. Set the size of the image. The options are explained here: https://www.w3schools.com/cssref/css3_pr_background-size.asp.
8. Set the position of the image in the box.
9. Finally, click **OK**.

> **Note**
>
> It is also possible to set an element's background in a style sheet; see "Styling templates with CSS files" on page 296. When referring to images or fonts from a CSS file, refer to a path that is relative to the current path, which is css/. For example: **#header { background-image: url('../images/image.jpg'); }**.

Border

In any type of template, boxes, tables and table cells, paragraphs, images and other elements can have a border.

Elements have a rectangular shape, so their border has four sides. Each side of the border can have a different layout.

Adding a border

1. Right-click the element and click the respective element on the shortcut menu. Alternatively, select the element (see "Selecting an element" on page 196) and on the **Format** menu click the respective element.
2. Click the **Border** tab.
3. Uncheck the option **Same for all sides** to be able to style each side of the border separately.
4. Specify the width of the border (side). This is equivalent to the `border-width` property in CSS.
5. Specify the style of the border (side), such as solid, dashed or dotted. This is equivalent to the `border-style` property in CSS.
6. Specify the color of the border (side): select a named color (defined in the "Colors Properties" on page 492) from the drop-down, or click the colored square to open the Color Picker dialog ("Color Picker" on page 492). Alternatively you could type a name or value in the Color field directly. It must be a predefined CSS color name (CSS color

names), a hexadecimal color code (HTML Hex Color), an RGB color value, for example `rgb(216,255,170)` or a CMYK color value, for example `cmyk(15%, 0%, 33%, 0%)`.
This setting is equivalent to the `border-color` property in CSS.

> **Note**
>
> It is also possible to set an element's border in a style sheet; see "Styling templates with CSS files" on page 296.

Rounding corners

Any element in a template can have rounded corners. For boxes and images, this option is available in the Formatting dialog. For other elements, you have to create a CSS rule to set the `border-radius` of the element (or class of elements).

Boxes, images and tables

To round the corners of a box, image or table:

1. Select a Box, Image or Table element (see "Selecting an element" on page 196) and on the **Format** menu click the respective element. Alternatively, right-click the element and click the respective element on the shortcut menu.
2. On the first tab in the Formatting dialog (the **Box**, **Image** or **Table** tab respectively) specify the **corner radius** in a measure (10mm, 5px, 0.5in) or percentage (0 - 90%).
3. For a Box or Image, click **Apply** to see the effect without closing the dialog or **OK** to close the dialog.

For a Table, you have to take yet another step. Tables can't have rounded corners and collapsed borders at the same time. All built-in table styles in the Designer have collapsed borders. For the rounded corners to show, you must create a CSS rule that sets the table's `border-collapse` property to `separate` instead of `collapse`.

1. Click the **Advanced** button at the bottom of the Formatting dialog.
2. Under **Property**, type **border-collapse**.
3. Under **Value**, type **separate**.

4. Add a padding to keep the table cells from sticking out of the rounded corners: under **Property** type **padding** and under **Value** type a measure for the padding.
5. Click OK, and click OK again to close the Formatting dialog.

If the table's rounded corners are still not (fully) visible, check the styles for table cells. Table cells can have their own background color and by that, hide the table's background color - including the rounded corners. Table cells can have rounded corners as well, just as any other elements; see below.

Other elements

To round the corners of elements other than boxes and images, or to have different roundings on different corners, you have to make use of the CSS property: `border-radius`; see https://www.w3schools.com/css/css3_borders.asp.

This is, for example, how you could round the corners of a paragraph:

1. Select the paragraph (see "Selecting an element" on page 196) and then select **Format > Paragraph** on the menu, or right-click the paragraph and select **Paragraph** on the shortcut menu.
2. Click the **Advanced** button at the bottom of the Formatting dialog.
3. Under **Property**, type **border-radius**.
4. Under **Value**, type the value of the corner radius in a measure (10mm, 5px, 0.5in) or percentage (0 - 90%).
5. Click OK, and click OK again to close the Formatting dialog.

Using a CSS file

Of course you could also add this rule to a CSS file; see "Styling templates with CSS files" on page 296. The following rule sets the border-radius of the corners of all paragraphs to 5 pixels: `p { border-radius: 5px; }`.

To make this rule apply to one specific paragraph, first give the paragraph an ID (select the paragraph and type the **ID**, for example **rounded**, on the **Attributes** pane). Then add the ID to the selector of the CSS rule, for example `p#rounded { border-radius: 5px; }`.

To make the CSS rule apply to a set of paragraphs with the same class, first give the paragraphs the same **class** (for example **rounded**). Then add that class to the selector of the CSS rule, for example `p.rounded { border-radius: 5px; }`.

Colors

Colors make an important contribution to the look and feel of your templates. This topic explains how to define and apply colors and how to keep them consistent in different output channels.

Defining colors, spot colors and tints

Color selectors, such as the drop-down list on the toolbar, initially contain a small set of colors. Add your own colors so that they can be used throughout the templates, in all contexts and in color selector dialogs as well as with their names in style rules (see "Styling and formatting" on page 294).

Defining colors

To define colors:

1. Select **Edit > Colors** on the menu.
2. Add a color. There are two ways to do this:
   - Click the **New** button (the green plus).
   - Select an existing color from the list and copy it using the **Duplicate** button . (The Filter drop-down limits the list to colors of a certain type.) Select the new color and click the **Edit** button .
3. In the Edit color dialog, type a name for the color (or let the Designer create a name based on the values that you select). The color's name can be used in style sheets. It should not contain spaces or special characters.

   Tip

   Working with style sheets? Choose a name that reflects the purpose of the color, rather than a name that describes the color. This way you won't have to change the color's name in the style sheets when you change the color.

4. Click **Color**. (Tint is used for transparent colors.)
5. Select the color type: **CMYK** or **RGB**.

The letters **CMYK** stand for Cyan (a greenish-blue color), Magenta (reddish-purple), Yellow and Key (black). In color printing, these are the usual primary colors.
RGB stands for Red, Green and Blue. In the RGB color model, red, green, and blue light are added together in various ways to reproduce a wide range of colors. This model is typically used for electronic devices.
For information about the **Spot color** and **Overprint** options see "Defining a spot color" below.

6. Drag the slider bars to set the values for the color and click OK or Apply.

Defining a spot color

A **spot color** is any color generated by an ink (pure or mixed). Note that spot colors can only be used on certain printers.
If your printer can use spot colors and you want a spot color to be used in a Print context, define the color as described above, making sure to:

- Match the color's **name** to that of the spot color used in the printer.
- Check the option **Spot color**.
- If applicable, check the **Overprint** option for this spot color. Overprinting refers to the process of printing one color on top of other colors. This is sometimes required, for example to deal with special print applications, such as applying UV ink or varnish to a certain area, or to avoid mis-registration when printing black on top of colored areas.

> **Note**
>
> **Black overprint** can be enabled for text smaller than a given size; see "Overprint and black overprint" on page 136.

Defining a tint

A tint is a transparent color, based on another color in the template. To define a tint:

1. Select **Edit > Colors** on the menu.
2. Click the **New** button (the green plus) to add the tint.
3. From the Type drop-down, select **Tint**.
4. In the Edit color dialog, type a name for the color (or let the Designer create a name based on the values that you select). The color's name can be used in style sheets. This name

should not contain spaces or special characters.

5. Select one of the existing colors in the template as the **Source** of the color. The tint or opacity will be applied to this color.
6. Check **Use opacity** if you want to set the Tint slider to use Opacity instead.
7. Use the slider to set the percentage of the tint or opacity, or type the percentage directly in the input box and finally click OK.

Applying a color

Colors can be applied to elements in your templates locally or through style sheets.

Using colors in style sheets

It is highly recommended to use style sheets in templates right from the start. Even more so if the communications are going to be output to different output channels, or if they consist of different sections (for example, a covering letter followed by a policy). Using CSS with templates allows a consistent look and feel to be applied. A style sheet can change the look of multiple elements, making it unnecessary to format each and every element in the template, time and again, when the company's layout preferences change. See "Styling templates with CSS files" on page 296.

In style sheets, you can color every type of element that has a CSS color property, such as **color**, **background-color** or **border-color**. Use the color's name as it is defined in the Designer, or any legal color value: a valid color name (see color names on w3schools), hexadecimal color code (see w3school's color picker), RGB color value, for example rgb (216,255,170) or CMYK color value, for example cmyk(15%, 0%, 33%, 0%).
The following CSS rule applies MyColor, which is a custom color (see "Defining colors, spot colors and tints" on page 328), to the text of all paragraphs:

```
p {
color: MyColor;
}
```

CMYK colors

You may use the custom cmyk() CSS function to assign a CMYK color to any element, or a series of elements. The following example assigns a steel blue color as a background for all H1 elements:

```
h1 {
background-color: cmyk(33%, 17%, 0%, 20%);
}
```

Coloring text

Instead of using a style sheet (see above), you can color text locally:

1. Select text or an HTML element that contains text (see "Selecting an element" on page 196).
2. On the menu, select **Format > Color**, or click the black triangle on the **Text color** toolbar button.
3. Select one of the colors in the list, or click **Other** to set all aspects of the text style, including text color and/or background color.

Coloring backgrounds and borders

Instead of using a style sheet (see above), you can color a background or border locally. This is how:

1. Select an HTML element (see "Selecting an element" on page 196).
2. On the **Format** menu, click the element. For a **div** element, click **Box**. The Formatting dialog opens up.
3. Click the **Border** or **Background** tab.
4. Click the downward pointing arrow next to **Color** to select a color from the list of predefined colors (see "Defining colors, spot colors and tints" on page 328).
   Alternatively, click the small rectangle to the right of the color list to open the Color Picker dialog. In this dialog you can select a color from the color wheel. You can also choose the color mode: RGB or CMYK. For an explanation of these two modes, see "Defining colors, spot colors and tints" on page 328; for an explanation of the other options in this dialog, see "Color Picker" on page 492.
   You could also type a name or value in the Color field directly. It must be a valid color name (see color names on w3schools), a hexadecimal color code (see w3school's color picker), RGB color value, for example `rgb(216,255,170)` or CMYK color value, for example `cmyk(15%, 0%, 33%, 0%)`.
5. Click **OK** or **Apply**.

Color management

Color profiles can keep colors consistent across different outputs. To manage color profiles, select **Edit > Color settings**; for an explanation of the options in the Color settings dialog, see "Color Settings" on page 494.

Fonts

In templates for personalized customer communications you can use the operating system's fonts, including imported fonts. There are two ways to incorporate remote or "web" fonts, as well; see "Using remote fonts" on page 335.

When you are using a font that is not installed on your machine (for example, the bold or italic variant of a regular font) Windows tries to simulate the font in the Designer. Likewise, PrintShop Mail Connect tries to simulate the font in the output. It is however not guaranteed that the output will be exactly as shown in the Designer. It is strongly advised to make sure that all used fonts are available and to always test the output before running production jobs.

> **Note**
>
> Hosting non-standard fonts on the operating system in a **server** environment (as opposed to importing them into the template) is not recommended.
> If output is produced on the server whilst running under a different account, that account might not have access to the font.
>
> If you do add a font to a server, select the option "Install for all users", and do not forget to **restart** the machine, as otherwise the font might not be available, due to the way certain Windows versions handle fonts.

Applying a font

To apply a particular font to a piece of text, you can:

- Select some text, or an element that contains text (see: "Selecting an element" on page 196) and select a font from the **Fonts** drop-down on the toolbar.
- Use the name of the font in a CSS rule, for example:
  ```
  body {
  font-family: Verdana, Arial, sans-serif;
  }
  ```
  Instead of the body tag, any element that can have the CSS property ‘font-family’ can be used.
  Make sure that the rule is applied to the text that you wanted to apply the font to; see "Step 2: apply CSS to the content" on page 302.

Note

The reason for specifying more than one font in a style sheet for emails is that the font might not be available on the device on which they are viewed.
Order the font names by preference. The last one should be the generic font family (either serif or sans-serif).

Importing a font

To import a font into a template:

- Drag the appropriate font files into the **Fonts** folder on the **Resources** pane.

Note

Font software may have specific restrictions for copying and redistribution. Please consult the license agreement for each font vendor before using it in a template. It is your responsibility to comply with the requirements of third-party agreements.

Note

Vertically-oriented fonts (fonts whose typeface name begin with the at (@) character) are not supported in Connect.

Using separate font files for font effects

Fonts normally come with separate files for the **bold**, *italic* and any other versions of the font. When you style text, Connect will use the appropriate font file to display the text, if that font file is linked to the respective font effect.
Here's how to combine a font file with a font effect.

1. Import the files for the bold, italic and any other versions of the font into the Fonts folder. Initially, imported fonts appear as separate entries in the Fonts drop-down on the toolbar. They are not used automatically when you style text, for example using the Bold and/or Italic toolbar button. For that, they have to be linked to a font effect.
2. Open the **Font Manager**: from the menu, select **Edit > Fonts**.

3. Combine each of the styled fonts with a font effect:
    a. Select the font and click the **Edit** button.
    b. Select the appropriate font effect (e.g. Font Weight: Bold, or Font Style: Italic).
    c. Change the name of the styled font. It should have the **same** name as the regular font.
4. Close the Font Manager.

Style simulation

When it can't use a separate font file for a particular font effect, Connect will **mimic** the bold and italic versions of that font. This is called style simulation.
There are some major drawbacks to style simulation. Firstly, it is not reliable: it does not work for all fonts. Secondly, style simulation is costly in terms of performance (output is created slower) and results in bigger files. Finally, because of the plain machine transformation of the font, the result will probably not look as good as when using a carefully designed, properly styled, font.
It is much better, if you have separate files for the **bold**, *italic* and any other versions of a font, to use those.

Font types

The Designer currently supports 5 font types: TTF, OTF, WOFF, EOT and SVG.

When creating an **Email** template, it's better to import several types of the same font, in order for any client to see the appropriate fonts.

In the case of a **Print** context you do not need to provide alternative fonts, because the output is not displayed using a font from the device on which the output is read.

Applying an imported font

Once a font is imported, it is automatically added to the Fonts drop-down on the toolbar.

It can also be used in the style sheets, even in combination with other fonts, for example:

```
body {
font-family: 'MyWebFont', Arial, sans-serif;
}
```

Using remote fonts

Adding remote web fonts directly

Release 2022.1 provides a straightforward method for using web fonts in Designer.

1. With Designer open, select the **Fonts** folder from the Resources tab. Right-click to display the context menu.
2. Select **Web Font**.
3. Provide a name for the font, and enter the URL.

This adds a link to the font in the Fonts folder, and the font can be used in your templates. Note that some email clients are not able to display web fonts used in email messages. In those scenarios the email will use the fall back font of the client (unless specified differently in a stylesheet.

When choosing fonts, we suggest adding multiple styles (regular, italic, bold weights etc). When dedicated font styles are not available, the browser has to synthesize styles such as bold and italic. You can pick multiple font families and still get one stylesheet URL. For example, fonts from Google Fonts include all the chosen fonts.

Using a style sheet to point to the font style sheet

This is the legacy method for adding web fonts to Designer. It is also supported, and templates created using this method will continue to work.

In order to use a remote font, you have to add a remote style sheet that points to a web font style sheet, for example https://fonts.googleapis.com/css?family=Roboto+Slab. For instructions see "Using a remote style sheet" on page 299.

Remote fonts can be applied to content in a Master page, section, or Snippet. They may be used in a style sheet and they are automatically added to the Fonts drop-down on the toolbar. Note that the list of font names is based on the style sheets that are included in the active section (see "Applying a style sheet to a section" on page 304) or, when editing a Snippet, in the section that was active when the Snippet was opened.

Note

Support for remote fonts in email clients cannot be relied upon, and not all remote fonts are supported by all browsers. It is therefore recommended to add fallback fonts to the specific style rules whenever using remote fonts in a **Web** or **Email** section (see "Applying a font" on page 332).

Locale

The locale is a setting that can affect date, time and currency output, and other formatting that depends on location and language. This setting is specific to each template, so changing it for one template will not affect other templates.

Assume that a record set has a `Date` field that contains the following date: `4/11/12`, and that this field has been added to the template using the Text Script Wizard with the Long Date format (see "Using the Text Script Wizard" on page 353 and "Formatting variable data" on page 357). If the locale is set to `en-US`, the date appears on the page as **April 1, 2016**. Setting the locale to `fr-CA` makes this text appear as **1 avril 2016**. Setting it to `zh-CN` will print **2016年4月1日** .

The locale can also be used in scripts; see "Writing your own scripts" on page 393 and "Standard Script API" on page 810.

Changing the locale

By default, the locale is the same as the operating system's locale setting. To change this setting for the currently open template:

1. On the menu, select **Edit > Locale**.
2. Use the drop-down to select how the locale is to be set for the current template:
    - Select **System Locale** to use the operating system's locale settings. The operating system's locale is set in the **Region** settings of the control panel. Note that when output is generated on a different operating system, that operating system's locale will be used.
    - Select **Explicit Locale** to specify a static locale which will remain static for this template, whichever server the template is used on. Use the **Locale** drop-down to select a specific locale. The locales comprise a language code followed by a 2-

letter country code (`de-DE`, `zh-CN`, `fr-CA`, `fr-FR`, etc), as defined by the international standards ISO-639-1 and ISO 3166.

- Select **Data Field** to use a data field from the record. The locale will be record-specific in this case. Use the drop-down to select a field within the current Data Model that contains the locale.
- Select **Parameter** to use the value of a runtime parameter. Runtime parameters are defined in the template, but their value is set at runtime; see Runtime parameters.

> **Note**
>
> The Data Field or Runtime Parameter value must be a string that contains the exact locale to be used, such as "en" or "fr-CA". It cannot be an alias such as "english" or "french". The locale supports language codes (`en`, `fr`, etc), as well as language codes followed by a hyphen and a 2-letter country code (`de-DE`, `zh-CN`, `fr-CA`, `fr-FR`, etc). The language codes are defined by ISO-639-1. The 2-letter country code is defined by ISO 3166. It is allowed to use an underscore instead of a hyphen (`de_DE`, `zh_CN`, `fr_CA`, `fr_FR`, etc).

3. Click **OK** to apply the setting. The setting will be saved with the template.

Spacing

Boxes, tables, paragraphs and many other elements have a **margin** and **padding**.
The margin is the white space around an element, outside the border. It is used to position an element in relation to the other elements, by putting more space between the element and its surrounding elements.
The padding is the space between an element's content and its border. It is used to position the content of the element inside the border.

Elements have a rectangular shape, so they have four sides. The margin and padding have be different on all sides.

> **Tip**
>
> Use a negative left margin to create a hanging paragraph or image.

To set the spacing:

1. Right-click the element and click the respective element on the shortcut menu. Alternatively, select the element (see "Selecting an element" on page 196) and on the **Format** menu click the respective element.
2. Click the **Spacing** tab.

> **Note**
>
> All settings in the Formatting dialog are in fact CSS style rules. Click the **Advanced** button to manually add CSS properties (at the left) and values (at the right). For more information about CSS, see "Styling and formatting" on page 294.
>
> It is also possible to change an element's formatting via a style sheet; see "Styling templates with CSS files" on page 296.

3. Set the value for the **padding** in measure or percentage. You can do this for each side separately, which is equivalent to the **padding-top**, **padding-bottom**, **padding-left** or **padding-right** property in CSS. To set the same padding for all sides, check the option **Same for all sides**. This is equivalent to the **padding** property in CSS.
4. Set the value for the margin in measure or percentage. You can do this for each side separately, which is equivalent to the **margin-top**, **margin-bottom**, **margin-left** or **margin-right** property in CSS. To set the same margin for all sides, check the option **Same for all sides**. This is equivalent to the **margin** property in CSS.
5. Click **OK**, or click **Apply** to apply the changes without closing the dialog.

Personalizing content

Variable-data printing is a form of digital printing in which elements such as text and graphics may be changed using information from a database or data file. It prints unique documents with customized messages for each customer. This is exactly what you can do with Connect: using variable data you can personalize your company's communications (including but not limited to printed matter).

Before you can start personalizing the content of a template, you must open a data file or database; see: "Loading data" on page 341.

The most common ways to personalize templates are listed below.

Variable data

Variable data are data from a database or data file that are used to personalize documents. Data fields can be inserted in the text directly. For example, if a person's last name can be found in your data, the field that holds the last name can be used in the text of a letter, invoice or email. Scripts in PrintShop Mail Connect Designer are the basis of Variable Data Publishing and are used to personalize documents based on your data.

The easiest, quickest and most direct way to add customer data to content is via drag and drop; see "Variable Data" on page 351.

The drag-and-drop method results in a Text Script. Another way to create a Text Script is to use the Text Script Wizard. Often it is better to use the Text Script Wizard than the drag-and-drop method.
The Text Script Wizard gives you more control over the way data is displayed. It can insert one or more data fields, each with an optional prefix and suffix. For blocks of data, such as addresses, the Text Script Wizard definitely is the better choice. See "Using the Text Script Wizard" on page 353.

Conditional content

In a template you may want to reveal content - text or images - to one group of recipients, but hide it from others. You can use a Conditional Script Wizard to achieve this, if you have a data field in your data on the basis of which a condition can be set. See "Showing content conditionally" on page 358.

Conditional Print sections

Entire Print sections can be included in or omitted from the output on the basis of one or more values in variable data. See "Conditional Print sections" on page 363.

Dynamic images and Print section backgrounds

Dynamic Images are dynamic in the sense that they are replaced by another image when a data field contains a certain value. Think of a signature image being swapped based on the sender's name, for example. You can use the Dynamic Image Script Wizard to make this happen; see "Dynamic images" on page 366.

To swap a Print section's background dynamically, you can use the Dynamic Background Script Wizard; see "Dynamic Print section backgrounds" on page 368.

Snippets

Snippets are pieces of content that can be re-used within the same template, in all contexts and sections. Snippets can contain any contents that a section can have, such as text, images, variable data, dynamic tables, etc. They are often very useful to personalize content, especially in combination with variable data and scripts. See "Snippets" on page 285 and "Loading a snippet via a script" on page 420.

Scripts

As soon as you want to do more than what can be done with the available Script Wizards, self-made scripts are the solution. You could, for example, combine data of two or more data fields in a condition for conditional text. Or you could load a part of a snippet depending on the value of a data field. With a self-made script you can achieve anything that can be done by any of the Script Wizards, and much more.

The basics of script-writing in the Designer are explained in the following topic: "Writing your own scripts" on page 393.

Control Scripts

When output is generated from a template, Control Scripts run **before** all other scripts, when a record is merged with a context. They determine how different sections of the context are handled. They can, for example, make the page numbering continue over all Print sections, split Email attachments, or omit Print sections from the output.

Some knowledge of JavaScript is needed to edit Control Scripts, just as for any other self-made scripts; see "Writing your own scripts" on page 393. Some functionality is provided as a Scripting Wizard, for example Conditional Print Sections.

See "Control Scripts" on page 429.

Post Pagination Scripts

Post Pagination Scripts are run in a **Print** context **after** the content has been paginated. Because they can search through the output of all Print sections, and modify Print sections (one

at a time), they may be used to create a Table Of Contents (TOC), as explained in the topic: "Creating a Table Of Contents" on page 447.

See "Post Pagination Scripts" on page 445.

Handlebars

Handlebars is the name of a JavaScript library that implements a templating language (see https://handlebarsjs.com/). It uses a template and an input object to generate output.

The Handlebars library is **integrated** in OL Connect Designer. This means that you can use Handlebars expressions in a special type of **snippets**: Handlebars templates.

See: "Handlebars in OL Connect" on page 452.

Loading data

Before you can add variable data fields to a template in the Designer, you need to have a Data Model and a sample of customer data. At the design stage the Designer doesn't need to have access to all data; it just needs to know which data fields exist in your data and it needs some data to be able to display a preview of the output.

To get access to a Data Model **and** data, you can open:

- A data file, see "Adding data from a data file" on the facing page.
- A database, see "Adding data from a database" on page 344.
- JSON sample data, see "Adding JSON sample data" on page 349.

When you open a data file or a database, the Data Model will be derived from it.

After opening a data file or database, the **Data Model** pane at the right hand bottom shows the data fields that occur in the data.

The **Value** column displays data from the first record in the data file. Use the **First**, **Previous**, **Next** and **Last** buttons to browse through the records, or use the Page Up, Page Down, Home and End keys.

Note that all values are strings.

> **Tip**
>
> If you have no data at hand, download a demo from http://demo.objectiflune.com and open a dummy data file to test with.

Adding data from a data file

1. Click **File**, select **Add Data** and then click **From file data source**. Browse to the location of the file and select it.
   The Designer can open the following types of data files:
   - Tabular files: CSV files (.csv, .txt) and Excel files (.xls, .xlsx)

     > **Note**
     >
     > Excel files saved in "Strict Open XML" format are not supported yet.

   - Microsoft Access Database (.mdb, .accddb)
   - JSON files (JSON)
   - XML files (.XML)
   - AFP files (optional).
2. Review the options presented, to ensure that the data will be interpreted correctly. The options available depend on the type of data file (see below).

Excel (XLS/XLSX) file options

- **First row contains field names**: Check this option to use the first line of the Excel file as headers. This option automatically names all extracted fields.
- **Sheet**: Only one sheet can be selected as the data source.
  - **Sort on**: Select a field on which to sort the data, in ascending (A-Z) or descending (Z-A) order. Note that sorting is always textual. Even if the selected column has numbers, it will be sorted as a text.

CSV file options

- **Encoding**: The Designer can not infer from a CSV file what encoding it is in. The default is right in the large majority of cases, but when it isn't, it can be very difficult to figure out the correct encoding. Ask your source what the encoding of the file is.
- **Field separator**: Choose the character that separates the fields in the file.
- **Comment delimiter**: If there are comment lines in the file, type the character that starts a comment line.
- **Text Delimiter**: Type the character that surrounds text fields in the file. Other delimiters will not be interpreted within these text delimiters.
- **Ignore unparsable lines**: When checked, any line that does not correspond to the above settings will be ignored.
- **First row contains field names**: Check this option to use the first line of the CSV as headers. This option automatically names all extracted fields.
    - **Sort on**: Select a field on which to sort the data, in ascending (A-Z) or descending (Z-A) order. Note that sorting is always textual. Even if the selected column has numbers, it will be sorted as a text.

MDB file options

- **File**: Include the full path to the file.
- **Password**: If the file isn't password protected, you can click **Next** without filling out this field.
- **Table name**: Use the drop-down to select the appropriate table or stored query to retrieve the appropriate data set.
- **Encoding**: Use the drop-down to select the encoding with which to read the data in the table.
    - **Sort on**: Select a field on which to sort the data, in ascending (A-Z) or descending (Z-A) order. Note that sorting is always textual. Even if the selected column has numbers, it will be sorted as a text.

JSON file options

After selecting a JSON file, specify if and how the JSON file must be split into multiple records. This is done by selecting an object or array as **parent element**. Its direct child elements - objects and arrays, *not* key-value pairs - can be seen as individual source records. If the root is selected, there will be only one source record.

Whether source records are output as individual records depends on the **trigger**. Either:

- Select **On element** to create a new record in the output for each object or array in the parent element.
- Select **On change** to create a new record each time the value in a certain key-value pair changes. Only key-value pairs that exist at the root of a child element can be evaluated.

All data found in child elements of the selected parent element are extracted to fields at the root of the Data Model. If a value consists of an object or array, the entire object or array is extracted to one data field.

Field names are derived from keys, objects and arrays in the *first* record, but those aren't necessarily the same in a subsequent record. If following records have a different structure, for example if a record has more child elements compared to the first record, some data may not get extracted.

> **Tip**
>
> JSON data can also be imported directly into the Data Model; see "Adding JSON sample data" on page 349.

XML file options

Select what level of XML elements defines a record.

The **Trigger** is what triggers the creation of a new record. It can be set to:

- **On element**: This defines a new record when a new element occurs on the selected XML level.
- **On change**: This defines a new record when a specific field under the chosen XML level has a new value. After selecting this option, you have to select the field that triggers the creation of a new record.

Adding data from a database

1. Click **File**, select **Add Data** and then click **From database data source**. Browse to the location of the file and select it.
   The Designer can open databases from the following types of data sources:

- MariaDB
- MySQL
- Microsoft Access Database (.mdb, .accddb)
- SQL Server
- ODBC DataSource
- JDBC
- Oracle.

2. Review the options presented. The options available depend on the type of database data source; see below.

MariaDB \ MySQL

1. Enter the appropriate information to connect to the database:
   - **Server**: Enter the server address for the MariaDB or MySQL database.
   - **Port**: Enter the port to communicate with the MariaDB or MySQL server. The default port is *3306*.
   - **Database name**: Enter the exact name of the database from where the data should be extracted.
   - **User name**: Enter a user name that has access to the MariaDB or MySQL server and specified database. The user only requires *Read* access to the database.
   - **Password**: Enter the password that matches the username above.
2. Click **Next** and enter the information for the source table.
   - **Connection string**: Displays the full path to the database.
   - **Table**: Use the drop-down to select the appropriate table or stored query to retrieve the appropriate data set.
   - **Encoding**: Use the drop-down to select the encoding with which to read the data in the table.
     - **Sort on**: Select a field on which to sort the data, in ascending (A-Z) or descending (Z-A) order. Note that sorting is always textual. Even if the selected column has numbers, it will be sorted as a text.
3. Click **Finish** to open the database.

Microsoft Access

1. Enter the appropriate information to connect to the database:
    - **File name**: Browse to your Microsoft Access database file (.mdb)
    - **Password**: Enter a password if one is required.
2. Click **Next** and enter the information for the source table.
    - **Connection string**: Displays the full path to the database.
    - **Table**: Use the drop-down to select the appropriate table or stored query to retrieve the appropriate data set.
    - **Encoding**: Use the drop-down to select the encoding with which to read the data in the table.
        - **Sort on**: Select a field on which to sort the data, in ascending (A-Z) or descending (Z-A) order. Note that sorting is always textual. Even if the selected column has numbers, it will be sorted as a text.
3. Click **Finish** to open the database.

SQL Server

1. Enter the appropriate information to connect to the database:
    - **Server**: Enter the server address for the SQLServer database.
    - **Port**: Enter the port to communicate with the SQLServer. The default port is *1433*.
    - **Database name**: Enter the exact name of the database from where the data should be extracted.
    - **User name**: Enter a user name that has access to the SQLServer and specified database. The user only requires *Read* access to the database.
    - **Password**: Enter the password that matches the user name above.
2. Click **Next** and enter the information for the source table.
    - **Connection string**: Displays the full path to the database.
    - **Table**: Use the drop-down to select the appropriate table or stored query to retrieve the appropriate data set.
    - **Encoding**: Use the drop-down to select the encoding with which to read the data in the table.
        - **Sort on**: Select a field on which to sort the data, in ascending (A-Z) or

descending (Z-A) order. Note that sorting is always textual. Even if the selected column has numbers, it will be sorted as a text.

3. Click **Finish** to open the database.

ODBC DataSource

1. Select the ODBC system data source. Note: Only 32-bit data sources are currently shown in this dialog, even if your system is 64-bits.
2. Click **Next** and enter the information for the source table.
    - **Connection string**: Displays the full path to the database.
    - **Table**: Use the drop-down to select the appropriate table or stored query to retrieve the appropriate data set.
    - **Encoding**: Use the drop-down to select the encoding with which to read the data in the table.
        - **Sort on**: Select a field on which to sort the data, in ascending (A-Z) or descending (Z-A) order. Note that sorting is always textual. Even if the selected column has numbers, it will be sorted as a text.
3. Click **Finish** to open the database

JDBC

1. Enter the appropriate information to connect to the database:
    - **JDBC Driver**: Use the drop-down to select which JDBC Driver to use for the database connection.
    - **JAR file path**: Enter a path to the JAR file that contains the appropriate driver for the database below.
    - **Server**: Enter the server address for the database server.
    - **Port**: Enter the port to communicate with the server.
    - **Database name**: Enter the exact name of the database from where the data should be extracted.
    - **User name**: Enter a username that has access to the server and specified database. The user only requires *Read* access to the database.
    - **Password**: Enter the password that matches the username above.
    - **Advanced mode**: check to enable the Connection String to manually enter the

database connection string.

- **Connection string**: Type or copy in your connection string.

2. Click Next and enter the information for the source table.
    - **Connection string**: Displays the full path to the database.
    - **Table**: Use the drop-down to select the appropriate table or stored query to retrieve the appropriate data set.
    - **Encoding**: Use the drop-down to select the encoding with which to read the data in the table.
        - **Sort on**: Select a field on which to sort the data, in ascending (A-Z) or descending (Z-A) order. Note that sorting is always textual. Even if the selected column has numbers, it will be sorted as a text.
3. Click **Finish** to open the database.

Oracle

1. Enter the appropriate information to connect to the database:
    - **Server**: Enter the server address for the Oracle database.
    - **Port**: Enter the port to communicate with the Oracle server.
    - **Database name**: Enter the exact name of the database from where the data should be extracted.
    - **User name**: Enter a username that has access to the Oracle server and specified database. The user only requires *Read* access to the database.
    - **Password**: Enter the password that matches the username above.
2. Click **Next** and enter the information for the source table.
    - **Connection string**: Displays the full path to the database.
    - **Table**: Use the drop-down to select the appropriate table or stored query to retrieve the appropriate data set.
    - **Encoding**: Use the drop-down to select the encoding with which to read the data in the table.
        - **Sort on**: Select a field on which to sort the data, in ascending (A-Z) or descending (Z-A) order. Note that sorting is always textual. Even if the selected column has numbers, it will be sorted as a text.
3. Click **Finish** to open the database.

After adding data from a database, the **Data Model** pane at the right hand bottom shows the data fields that occur in the data.

The **Value** column displays data from the first record in the data file. Use the **First**, **Previous**, **Next** and **Last** buttons to browse through the records.

Adding JSON sample data

JSON data can either be **added to**, or **replace** the Data Model in a template.

To add JSON sample data to the Data Model:

1. Select **File > Add Data > JSON sample data**, from the menu. Alternatively, click the **JSON Sample Data** toolbar button on the Data Model pane.
2. Either browse to the location of the file and select it, or paste or write the JSON directly in the box below the file name.

> **Tip**
>
> If a data field contains JSON data you could use that data as sample data: right-click the field, select **Copy** to copy the JSON data to the clipboard; then open the JSON Sample Data dialog and paste the data there.

3. Review the JSON; you may edit it if you like.
4. Select the **Replace Data Model** option if you want the JSON to replace the existing Data Model. Otherwise, the JSON data will be mapped to corresponding fields in the existing Data Model, and data that cannot be mapped to any field will be discarded.
5. Click **Finish**.

> **Note**
>
> Values are converted to text strings. If a value in a record object is another JSON object, it is ignored.

Add a counter using the Generate Counter Wizard

Generating a counter is useful for numbered tickets or any other template requiring sequential numbers but no variable data.

The Generate Counter Wizard creates a record set with a Counter field and in that field, the current counter value for each record. The Counter starts and stops at set values and is incremented by a set value as well.

1. To open the Generate Counter Wizard, select **File > Add data > Generate counters**.
2. Adjust the settings:
    - **Starting value**: The starting number for the counter. Defaults to 1.
    - **Increment value**: The value by which to increment the counter for each record. For example, an increment value of 3 and starting value of 1 would give the counter values of 1, 4, 7, 10, [...]
    - **Number of records**: The total number of counter records to generate. This is not the end value but rather the total number of actual records to generate.
    - **Padding character**: Which character to add if the counter's value is smaller than the width.
    - **Width**: The number of digits the counter will have (prefix and suffix not included). If the width is larger than the current counter value, the padding character will be used on the left of the counter value, until the width is equal to the set value. For example for a counter value of "15", a width of "4" and padding character of "0", the value will become "0015".
    - **Prefix**: String to add before the counter, for example, adding # to get #00001. The prefix length is not counted in the width.
    - **Suffix**: String to add after the counter. The suffix length is not counted in the width.
3. Click **Finish** to generate the Counter record set.

Tip

While the Generate Counter script is really useful for things like raffle tickets, it's unusable in combination with a data file or database, as it cannot complement that data automatically. This can only be done with a script. A script that adds a counter to data, using the current record index to calculate the current counter value, can be found in this how-to: Manual counter in designer.

Variable Data

Variable data are data from a database or data file that are used to personalize documents. Data fields can be inserted in the text directly. For example, if a person's last name can be found in your data, the field that holds the last name can be used in the text of a letter, invoice or email. Scripts in PrintShop Mail Connect Designer are the basis of Variable Data Publishing and are used to personalize documents based on your data.

After loading data from a data file or database (see "Loading data" on page 341), you can add variable data fields to the contents of your template. You can do this via the drag-and-drop method, or using the Text Script Wizard.

Use the **Text Script Wizard** when there are empty fields in the data, and the value of a data field needs to be preceded or followed by a space, line break or text in the template. Otherwise, empty data fields will cause empty lines and superfluous white spaces to show up in the text. You should also use this method for blocks of data, such as address blocks, and when you want to format data differently, for example, when you want a number to be displayed as a currency.

You can use the **drag-and-drop** method for simple fields that do not need to be preceded or followed by a space, line break or text.

Inserting variable data directly (drag-and-drop)

An easy, quick and direct way to insert variable data in the content is via drag and drop.

1. Open the section you want to add the data field to.
2. Drag and drop a data field from the **Data Model** pane at the bottom right into the content of your template.
   To select and insert multiple data fields at the same time, press **Shift** or **Ctrl**, whilst selecting fields in the **Data Model** pane.

What happens is that:

- A **placeholder** for the value of the data field shows up in the text. It looks as follows: @FIELDNAME@.
- A **script** appears in the **Scripts** pane at the bottom left.

The script replaces the placeholders in the content with the value of a data field in the current record.

Switch to the **Preview** tab at the bottom of the workspace to see the script in operation. The value of the corresponding data field in the first record appears instead of the placeholder, everywhere where the placeholder is found in the text. This value will be refreshed when you browse through the records in the Data Model pane.
When the output (the letter, email, etc.) is generated, the text script executes for each record in the record set, and each time it replaces the placeholders by the value of the field in the current record.

In the **Scripts** pane you can see that the script has a **name** and a **selector**.

The drag-and-drop method automatically generates a script that is named after the data field (see the first column of the **Scripts** pane).

The **selector** (in the second column in the **Scripts** pane) is what a script looks for in the template. By default the drag-and-drop method wraps the placeholder text in a Span. (Select the placeholder and switch to the **Source** tab to see the Span element.) The Span's **class** is used as selector. Note that any spaces that occur in a data field name will be removed from the script selector.

When you drag the same field from the Data Model to the content again, a second placeholder appears in the text, but no new script is added. The existing script will find and replace all placeholders that match its selector.
You could also drag a script with an ID and/or class selector from the Scripts pane into the template to insert an additional placeholder.

Tip

Press the **CTRL** key while dragging to wrap the placeholder in an absolute positioned box (a **div**) at the cursor position. Each placeholder inside the box - there can be multiple if you drag multiple fields - is wrapped in a Span, and each Span gets its own class.
This method is particularly useful when the document mainly consists of a PDF used as the background image of a section (see "Using a PDF file or other image as background" on page 142).

Tip

Drag the data field directly to the **Scripts** pane to create a script with a class selector, without adding a placeholder to the template. Hold the Alt key while dragging to create a script with a text selector.

Note

Connect versions older than 2020.2 used the placeholder text as the script's selector. Looking for text in a text is a less optimized operation and may impact output speeds in longer documents. For that reason placeholders are now by default wrapped in a Span and the class of that Span is used as the script's selector.
If you would like the drag-and-drop method to behave like it did in older Connect versions, you can change its behavior via the Preferences (**Window > Preferences > Editing**), or hold the **ALT** key while dragging to change its behavior on the fly.

For more tips to make a template generate output faster, see "Optimizing scripts" on page 408.

Using the Text Script Wizard

The Text Script Wizard can insert one or more data fields into your template, each with an optional prefix and suffix. It is recommended to use the Text Script Wizard for blocks of data, such as address blocks, and when data fields can be empty or need to be formatted differently.

1. Create a new text script and open the Text Script Wizard. There are two ways to do this:
   - On the **Scripts** pane at the bottom left, click the black triangle on the **New** button and click **New Text Script**. A new script appears in the list. Double-click the new script to open it.
   - Select a word in the content. Right-click the selection and on the shortcut menu, choose **Text Script**.

   The Text Script Wizard appears.
2. Change the name of the script to make clear what it does.

> **Note**
>
> Scripts can only have the same name when they are not in the same folder.

3. The **selector** states the text to be found in the template. The results can be replaced by the script.

> **Tip**
>
> Hover over the name of a script in the **Scripts** pane to highlight parts of the template that are affected by the script.

- **Text**, for example: @lastname@, or {sender}. The text doesn't have to have any special characters, but special characters do make it easier to recognize the text for yourself. In the Text Script Wizard, click **Text** and type the text to find.

  > **Note**
  >
  > A script made with the Text Script Wizard for a block of data already runs faster than a series of individual scripts, because it only has one selector. However, searching for text can be a lengthy operation, compared to searching for an element with an ID. When speed matters, select one of the two remaining options: **Selector** or **Selector and Text**. See also: "Testing scripts" on page 403 and "Optimizing scripts" on page 408.

- An **HTML/CSS selector**:
  - HTML elements, such as a paragraph. In the Text Script Wizard, click **Selector** and type the HTML tag without the angle brackets, for example: **p**.
  - HTML elements with a specific class. In the Text Script Wizard, click **Selector** and type the class name, including the preceding dot, for example: **p.green** for all paragraphs with the class 'green' or **.green** for all kinds of HTML elements that have the class 'green'. See "Styling and formatting" on page 294 for an explanation about CSS (Cascading Style Sheets).

- An HTML element with a specific ID. In the Script Wizard, click **Selector** and type the ID, including the preceding #, for example: **#intro**.

  > **Note**
  >
  > Each ID should be unique. An ID can be used once in each section.

- Etcetera. See https://www.w3schools.com/cssref/css_selectors.asp for more selectors and combinations of selectors.

- A **selector and text**. This is text inside an HTML element (or several HTML elements) with a specific HTML tag, class or ID. In the Text Script Wizard, click **Selector and text** and type the selector and the text in the respective fields.

4. Click the the downward pointing arrow in the first row in the column **Field**. Select a data field from the list that appears.
5. Add a **Prefix** and/or a **Suffix**. The prefix and suffix can contain text and/or HTML tags. If a field is empty, the prefix and suffix will be ignored, which means you can add line returns and static text, such as:
   - with a Number field, Prefix: Your invoice (one space at the end), Suffix: is now ready to be viewed!
   - with a field LastName, Suffix
 (which adds a line break)
   - with a field State, Prefix: , (comma then space).

   For a comma between fields, use the Prefix of the second field, if you don't want a comma when the second field has no value.
6. The Wizard allows you to reformat the data (for example, apply uppercase, apply thousand separators to numbers, etc.). Click the column **Format**, click the downward pointing arrow and select one of the formats. For an explanation of the formats see "Formatting variable data" on page 357.
7. Add as many data fields as you need, following the same procedure.
8. Optionally, you can click **Options** to specify where and how the script inserts its results:
   - As **HTML**. HTML elements in the results are processed and displayed as HTML elements. For instance, `<b>this is bold</b>` will be displayed as **this is bold**. This is the default setting.
   - As **text**. This inserts the results as-is, meaning HTML tags and elements are displayed as text in the output. In this scenario, "<br>" shows up in the text and does

not insert a line break.

- As the value of an **attribute** of an HTML element. The selector of the script should be an HTML element. Which attributes are available depends on the selected HTML element. If the script's selector is an image (<img> element) for example, and the attribute is `src`, the script will modify the image's source. The script's results should be a valid value for the chosen attribute.

> **Note**
>
> When checked, the option **Convert fields to JSON string** writes the results from the script into an attribute or text as a JSON string. This is useful for web contexts where a front-end script can read this value easily.

9. Close the Text Script Wizard and type the placeholder for the results of the script in the content of your template, or make sure that there is at least one element that matches the selector of the script.
10. Hover over the name of the script in the **Scripts** pane. In the workspace you will see which parts of the template are affected by the script. If the script produces an error, the error message will be displayed in a hint on the **Scripts** pane.

> **Tip**
>
> When one of the included data fields is empty, the respective line, including the prefix and suffix, is skipped. The result of the script will be shorter, causing the rest of the content to move up or down. If, in a Print context, you don't want the result of the script to be part of the text flow (for example, when a letter is going to be sent in an envelope with a window), put the placeholder for the script in a positioned box (see "Boxes" on page 256 and "How to position elements" on page 312).

> **Tip**
>
> - An example of how to create an address block using the Text Script Wizard is described in a how-to; see How to create an Address Block.

- To use only part of a data field, or to split the data, you will have to write a script. For an example, see this How-to: How to split a string into elements.

Formatting variable data

When a Text Script, made with the Text Script Wizard (see "Using the Text Script Wizard" on page 353) adds variable data to a template, it can easily change the way the data are formatted as well. This is done in the Text Script Wizard through a special formatting modifier or a format mask for each field that the script adds to the template.

Note

The locale influences the way dates, times, numbers and currencies are formatted; see "Locale" on page 336.

Formatting can be applied to **values in a Dynamic Table** via the `data-format` attribute. See: Formatting values in a Dynamic Table.

You could also format data in a **script** using the `formatter`; see "Standard Script API" on page 810.

Date

Dates can only be formatted via a script. All data in the Data Model are strings; they have to be converted to Date objects before applying a particular date format. See also: "Creating a Date object from a string" on page 821.

Font style

Text originating from variable data can be displayed in uppercase, lowercase or proper case.

1. Open the Text Script Wizard: double-click to open an existing script in the Scripts pane or create a new Text Script using the Text Script Wizard; see "Using the Text Script Wizard" on page 353.
2. Click a data field that contains text, or add such a data field to the script with the Add field button on the right.

3. Under **Format** choose the correct setting:
    - **Uppercase** transforms all characters to uppercase.
    - **Lowercase** displays transforms all characters to lowercase.
    - **Propercase** transforms the first character of each word to uppercase and all other characters to lowercase.
    - **None** leaves the text as is.
4. Close the Script Wizard. For a new script, don't forget to add the selector to the template.

Numbers and currencies

Numbers, and strings existing of digits, can be displayed as a number with a certain formatting or as an amount of money.

1. Open the Script Wizard: in the Scripts pane, double-click the script, or create a new Text Script using the Text Script Wizard; see "Using the Text Script Wizard" on page 353.
2. Click the data field that contains the numeric value that you want to display differently, or add the data field to the script with the Add field button on the right.
3. Under **Format** choose one of the following settings:
    - **Custom Pattern**: allows you to enter a custom format mask. For example, the pattern `000000` means that the number should count six digits; leading zeros are added to numbers shorter than six digits. For an overview of pattern symbols see "Number patterns" on page 844 and http://docs.oracle.com/javase/7/docs/api/java/text/DecimalFormat.html. The custom pattern will only work if the actual value of the field can be converted from a text to a number.
    - **Grouped** displays a number with three decimal places and sets the thousands separator for the value based on the current locale; see "Locale" on page 336.
    - **Currency** displays a number as an amount of money, with a thousands separator and rounded to two decimal places, based on the current locale; see "Locale" on page 336.
    - **Currency no symbol** does the same as Currency, but omits the currency symbol.
4. Close the Script Wizard. For a new script, don't forget to add the selector to the template.

Showing content conditionally

One way to personalize content is to show or hide one or more elements depending on the value of a data field or runtime parameter. For example, a paragraph written for Canadian

customers could be hidden when the recipient of the letter is not living in Canada, if that can be derived from the data.

Showing or hiding elements using the Conditional Content Script wizard

The Conditional Script Wizard helps you to show or hide an element – a paragraph, image or other HTML element - based on the value of one or more data fields and runtime parameters. For example, you could check whether the data field 'State' is 'Equal To' the value 'British Columbia' or 'Québec', to include a paragraph only for recipients in those states.

To open the Conditional Script wizard:

- Select one or more elements, right-click and choose **Make Conditional**.
- Alternatively, click the black triangle on the **New** button on the **Scripts** pane at the bottom left of the window, and click **Conditional Content Script**.

In the wizard, first make the following **settings**.

1. **Rename** the script so that it reflects what the script does.

   > **Note**
   >
   > Scripts can only have the same name when they are not in the same folder. (See "Managing scripts" on page 400.)

2. Type a **Selector**. The selector selects one or more pieces of text or elements from the template, so that the conditional content script can hide or show those pieces.
   An **ID** (for example: #conditional-script) is best if you want to show or hide one element only.
   Use a **class** selector (for example: .conditional) if the script should show or hide more than one element. See "Using the Text Script Wizard" on page 353 for further explanation on selectors.
   When you start the Conditional Script Wizard by right-clicking an element, the Selector field is pre-populated with the element's ID and class (or classes). If the element didn't have an ID or class, it gets a new ID (or class, in case multiple elements were selected) that is generated automatically.
   If you'd want to change the selector later, you can do that after reopening the script (double-click the name of the script in the **Scripts** pane).

3. Set the **Action**: use the drop-down to select whether to **Show** or **Hide** the element when the condition below is **true**. When the condition evaluates to false, the opposite action is performed.

Once these settings have been made, the **condition** can be build.

The **Add** button adds a rule that evaluates a **data field** to a group.
To add either a **group**, or a rule that evaluates a **runtime parameter**, click the downward pointing arrow next to this button and select **Group** or **Parameter Rule**.

> **Note**
>
> Creating a parameter rule is only possible if the template contains runtime parameters; see Runtime parameters.

The rule(s) apply at a Group level. There may be one rule or many rules at the same level, and there may be groups within groups, providing the ability to create quite complex nested logical structures.

Group

A group consists of one or more rules with a logic operator. Four logic choices are available at the Group level. These are:

- **All of the following**.
  This equates to the logical operator (... AND ...).
  If *all* of the associated criteria are met, then this group resolves to TRUE.
- **Any of the following**.
  This equates to the logical operator (... OR ...).
  If *any* of the associated criteria are met, then this group resolves to TRUE.
- **Not all of the following**.
  This equates to the logical operator (NOT(... AND ...)).
  If *any (but not all)* of the associated criteria are met, then this group resolves to TRUE.
- **Not any of the following**.
  This equates to the logical operator (NOT (... OR ...)).
  If *none* of the associated criteria are met, then this group resolves to TRUE.

The top level is always a group. A group can contain one or more Rules and/or Groups.

Rule

A single rule evaluates one data field or runtime parameter using a single operator or function and a value (the "operand").

To construct a rule:

1. Click the downward pointing arrow next to **Field**, then select the data field or runtime parameter that should be evaluated.
2. Select the **operator**. The downward pointing arrow expands the list of operators with which the data can be evaluated.
   The options are: *Equal to*, *Not equal to*, *Contains*, *Does not contain*, *Begins with*, *Ends with*.
3. Enter the **Value** that should be used for the conditional check.
   - Values are **case sensitive** by default; you can click the button next to the value to make them case insensitive.

It isn't possible to enter more than one value in one rule. Comparisons to multiple values can be made by combining rules in a group.

To save the script, click **Apply** or **OK**.

The selected action will be performed if the joint groups and rules evaluate to **true**.
If, conversely, the condition evaluates to false, the opposite action will be performed.

To see the result, toggle to the **Preview** tab at the bottom of the workspace (or select **View > Preview View** on the menu).

> **Note**
>
> If a condition cannot be made with the wizard - for example, one that compares a data field to a runtime parameter value - you can click **Expand** and edit the code of the script (see "Writing your own scripts" on page 393). But be careful: once the script has been altered, there is no going back to the wizard.

Showing or hiding several elements with one conditional script

To apply one conditional content script to several elements, a CSS **class** or HTML element needs to be used as the selector of the script. When you select multiple elements and create a new conditional content script following the actions described in "Showing or hiding elements using the Conditional Content Script wizard" on page 359, a CSS class will be applied automatically.

If you have created the script yourself, or if you want a script to apply to more elements later on, you have to do this manually.

1. Double-click the conditional script in the **Scripts** pane to reopen it.
2. Make sure that the selector is a CSS class (for example, .male) or an HTML element with a certain CSS class (for example, p.male). See "Using the Text Script Wizard" on page 353 for further explanation on selectors.
3. Apply the same CSS class to all elements that should be shown or hidden under the condition that you have set in the conditional script. Click each element and type the class (without the preceding dot) in the **Class** field.

Showing or hiding a text selection

When you right-click on an element and make it conditional, the element as a whole will be made conditional. This happens even when you select a few words in a paragraph and right-click those words; the paragraph as a whole will be made conditional.
It is, however, possible to show or hide a part of a paragraph.

If the text that you want to show or hide is short, you could use the **Selector and Text** option in the Conditional Content wizard and copy the text into the **Text** field. This option uses the specified selector as a trigger for the script. The script applies to all instances of the text found within the specified selector.

Searching for text can be a lengthy operation, compared to searching for an element with an ID. If performance is an issue, of if the search text is long, a better alternative is to wrap the text in a span element first.

1. Select the part of the text that you want to make conditional.
2. Right-click the selected text and click **Wrap in span**.

3. Type an **ID** and/or a **class**. An ID is fine if this is the only thing that should be shown or hidden on a given condition. Use a class if there is more that should be shown or hidden on the same condition.
4. Start creating a conditional content script from the **Scripts** pane. Use the ID or class as the selector of the script. See "Showing or hiding elements using the Conditional Content Script wizard" on page 359.

Conditional Print sections

You can include or exclude entire Print sections from the output, depending on a field's value. This can be done using the Conditional Print Section Script Wizard, described below. Alternatively you could write a Control Script (see "Control Scripts" on page 429).

Including or excluding Print sections using the Conditional Print Section Script wizard

To open the Conditional Print Section Script wizard:

- Right-click the section and click **Make Conditional**.
- Alternatively click the black triangle on the **New** button on the **Scripts** pane at the bottom left of the window, and click **Conditional Print Section Script**. Double-click the new script to open the Conditional Print Section Script wizard.

In the wizard, first make the following **settings**.

1. **Rename** the script so that it reflects what the script does.

> **Note**
>
> Scripts can only have the same name when they are not in the same folder. (See "Managing scripts" on page 400.)

2. Select the section you want to put a condition on.
3. Set the **Action**: **Print** or **Skip** that is performed when the condition below is **true**. The opposite action is applied when the condition returns false.

Once these settings have been made, the **condition** can be build.

The **Add** button adds a rule that evaluates a **data field** to a group.
To add either a **group**, or a rule that evaluates a **runtime parameter**, click the downward pointing arrow next to this button and select **Group** or **Parameter Rule**.

> **Note**
>
> Creating a parameter rule is only possible if the template contains runtime parameters; see Runtime parameters.

The rule(s) apply at a Group level. There may be one rule or many rules at the same level, and there may be groups within groups, providing the ability to create quite complex nested logical structures.

Group

A group consists of one or more rules with a logic operator. Four logic choices are available at the Group level. These are:

- **All of the following**.
  This equates to the logical operator (... AND ...).
  If *all* of the associated criteria are met, then this group resolves to TRUE.
- **Any of the following**.
  This equates to the logical operator (... OR ...).
  If *any* of the associated criteria are met, then this group resolves to TRUE.
- **Not all of the following**.
  This equates to the logical operator (NOT(... AND ...)).
  If *any (but not all)* of the associated criteria are met, then this group resolves to TRUE.
- **Not any of the following**.
  This equates to the logical operator (NOT (... OR ...)).
  If *none* of the associated criteria are met, then this group resolves to TRUE.

The top level is always a group. A group can contain one or more Rules and/or Groups.

Rule

A single rule evaluates one data field or runtime parameter using a single operator or function and a value (the "operand").

To construct a rule:

1. Click the downward pointing arrow next to **Field**, then select the data field or runtime parameter that should be evaluated.
2. Select the **operator**. The downward pointing arrow expands the list of operators with which the data can be evaluated.
   The options are: *Equal to*, *Not equal to*, *Contains*, *Does not contain*, *Begins with*, *Ends with*.
3. Enter the **Value** that should be used for the conditional check.
   - Values are **case sensitive** by default; you can click the button next to the value to make them case insensitive.

It isn't possible to enter more than one value in one rule. Comparisons to multiple values can be made by combining rules in a group.

To save the script, click **Apply** or **OK**.

The selected action will be performed if the joint groups and rules evaluate to **true**.
If, conversely, the condition evaluates to false, the opposite action will be performed.

To see the result, toggle to the **Preview** tab at the bottom of the workspace (or select **View > Preview View** on the menu).

Note

If a condition cannot be made with the wizard - for example, one that compares a data field to a runtime parameter value - you can click **Expand** and edit the code of the script (see "Writing your own scripts" on page 393). But be careful: once the script has been altered, there is no going back to the wizard.

Tip

Take a look at the Resources pane: on each Print section that is affected by a Conditional Print Section script a small decorator appears if it is skipped with the current data record.

Dynamic images

Dynamic images are called dynamic because they are switched, depending on the value of a data field. This way, a template can be adjusted to different customers.

This page describes how to insert dynamic images using the Dynamic Image Script Wizard.

Otherwise, adding dynamic images requires a self-made script (see "Writing your own scripts" on page 393, "template" on page 960 and "ImageResource" on page 965).

Adding dynamic images

Dynamic images can be added to the template using the Dynamic Image Script Wizard only if you have:

- One or more data fields that contain values on the basis of which the images can be switched.
- An appropriate image for each case. All files should be of the same type and they need to be stored in the same location (the **Images** folder on the **Resources** pane, a folder on disk, or online). It is important that they are named after the various possible values of the related data field.

To use the Dynamic Image Script Wizard:

1. Add one image to the template. See "Adding images" on page 273.
2. Right-click the image and click **Dynamic Image**. Or select the image and click **Source** (not the field, but the label before the field) in the Attributes pane.
   The Dynamic Image Script Wizard opens. The image's ID is used as the script's selector. If the image did not have an ID, it is automatically generated.
3. Select the location of the files:
   - Select **Resources** if the images reside in the **Images** folder on the **Resources** pane.
   - **Folder on disk** refers to a folder on a hard drive that is accessible from your computer. Click the **Browse** button to select a folder.
     As an alternative it is possible to enter the path manually. You can give a local path (e.g. C:\Images\) or use the "file" protocol. The complete syntax of a fully qualified URL with the "file" protocol is: `file://<host>/<path>`. Note: if the host

is "localhost", it can be omitted, resulting in `file:///<path>`, for example: `file:///c:/resources/images/`.

- **Online** requires you to give a specific web address (for example, http://www.mysite.com/images/).

4. The Dynamic Image Script Wizard composes the file names (including the path) based on the selected **location**, the **prefix**, the value of a **data field**, and the **suffix**. The variable part of the file names is the value of the data field(s) in the **Field** column. The prefix and suffix are meant to contain static parts of the file names.
   Click the first field in the column **Field**, and then click the downward pointing arrow.
   Select the data field to be evaluated.
   If you want the file name to be composed of the value of several data fields, simply click in the next row or click the **Add** button. This adds a row. Note that the rows will be concatenated to compose one file name. Only the last suffix should contain the file extension.
5. The resulting file name, including the path and file extension, is assigned to the **src** (source) attribute of the image. You can click **Options** to verify this.
6. Click **Apply** or **OK**. Now click the **Preview** tab and browse through the records to verify that the script works as expected.

Tip

The dynamic images feature can be used to insert dynamic signatures.

How to insert dynamic images if there are no data fields with the actual names of the images is described in another how-to: Dynamic image that doesn't contain the data field value.

Note

An image with an unknown file extension is represented by a red cross in the output, but no error is logged unless the image refers to a local file that does not exist on disk.
Image file extensions that the software recognizes are: AFP, BMP, EPS, GIF, JPG/JPEG, PCL, PDF, PNG, PS, SVG, and TIF/TIFF.

Editing a Dynamic Image

To edit dynamic images added to the template earlier, right-click the image, or the space reserved for the dynamic images. Then click **Dynamic Image** to open the Dynamic Image Script Wizard again.
Alternatively you may double-click the respective script on the Scripts pane.

Dynamic Print section backgrounds

Print sections can have a PDF background image (see "Using a PDF file or other image as background" on page 142). This feature is often used to create a Print template from a PDF file while making some minimal changes to it. Of course, the PDF file itself can't be edited in a Designer template, but when it is used as a section's background, text and other elements, such as a barcode, can be laid on top of it.

Dynamic backgrounds are switched automatically depending on the value of a data field.

Adding a dynamic background

Dynamic background PDFs can be added to a Print section using the Dynamic Section Background Script Wizard **if**:

- One or more data fields contain values on the basis of which the PDFs can be switched.
- There is an appropriate PDF for each case. It is important that the PDFs are named after the various possible values of the related data field.
- All PDFs are stored in one folder (the **Images** folder on the **Resources** pane, or an external folder).

Otherwise, adding a dynamic background requires a self-made script (for help on this, see "Control Script: Setting a Print section's background" on page 438).

To use the Dynamic Section Background Script Wizard:

1. Open the Wizard. There are three ways to do that:
    - Right-click the Print section on the **Resources** pane and click **Dynamic Background**.
    - On the **Scripts** pane, click the arrow next to the **New** button and select **Dynamic Background Script**. Double-click the script to open it.

- On the **Scripts** pane, right-click **Control** and select **New > Dynamic Background Script**.

2. Select the desired section from the **Section** drop-down. You may also change the name of the script.
3. Select the location of the files:
   - Select **Resources** if the PDFs reside in the **Images** folder on the **Resources** pane.
   - **Folder on disk** refers to a folder on a hard drive that is accessible from your computer. Click the **Browse** button to select a folder.
     As an alternative it is possible to enter the path manually. You can give a local path (e.g. C:\Images\) or use the "file" protocol. The complete syntax of a fully qualified URL with the "file" protocol is: `file://<host>/<path>`. Note: if the host is "localhost", it can be omitted, resulting in `file:///<path>`, for example: `file:///c:/resources/images/`.
   - **Online** requires you to give a specific web address (for example, http://www.mysite.com/images/).
4. The Dynamic Section Background Script Wizard composes the file names (including the path) based on the selected **location**, the **prefix**, the value of a **data field**, and the **suffix**. The variable part of the file names is the value of the data field(s) in the **Field** column. The prefix and suffix are meant to contain static parts of the file names.
   Click the first field in the column **Field**, and then click the downward pointing arrow. Select the data field to be evaluated.
   If you want the file name to be composed of the value of several data fields, simply click in the next row or click the **Add** button. This adds a row. Note that the rows will be concatenated to compose one file name. Only the last suffix should contain the file extension (.pdf).
5. Click **Apply** or **OK**.
   The script assigns the resulting file name, including the path and file extension, to the URL of the section's background.
6. Now click the **Preview** tab and browse through the records to verify that the script works as expected.

Tip

If you want to see the underlying script, reopen the script and click the **Expand** button. For more information on this type of script, see "Control Script: Setting a Print section's background" on page 438).

Preferences

The Preferences dialog is used to modify the general software preferences. Changes made in this dialog affect the software globally, not individual templates and data mapping configurations.

The Preferences dialog is separated into individual tabs, where each tab controls certain aspects of the software.

To open the Preferences dialog, select **Window > Preferences**.

The General preferences also provides you with buttons to :

- **Reset all Warning Dialogs**: This re-enables all warning dialogs that might have been previously disabled by selecting the "Don't show again" checkbox within the dialog.
- **Restore Defaults**. This option restores the preferences to Defaults. This applies to the current Preferences page only, but not other Preferences.
- **Apply**: This option applies the settings made within the current Preferences page, but does not close the Preferences dialog.

Clean-up Service preferences

The Clean-up Service defines how the Connect database and the temporary files created during Connect production runs are cleaned up after the production run has finished.

As part of the job production process PrintShop Mail Connect uses a database for intermediate storage and also creates various temporary "managed" files. These files include data extractions, configuration files and any intermediate files created during the production process. Connect keeps track of all these files through references held within the Connect database.

All the files created and the database references to them are stored for a set amount of time in order to allow Connect to reuse them. However, we do not want to store these indefinitely, because the database would run out of space. The solution is to use the "Clean-up Service" to remove the temporary data and files once they are no longer needed.

The more items that are present in the database, and the larger they are, the more time and processing power (CPU) that will be required for cleaning them up. Thus a regular Clean-up of the database (as often as possible) is recommended.

This is especially the case if items are not going to be retrieved from the database at a later date. i.e. If the Connect job is not going to be re-run.

The clean-up can always be set to run outside of business hours (see the "Run according to the cron schedule: Enter the interval at which the Clean-up service runs. To understand how to write a cron job schedule, please refer to the Quartz Scheduler tutorial: http://www.quartz-scheduler.org/documentation/quartz-2.3.0/tutorials/tutorial-lesson-06.html." below option below), to reduce impact upon Production systems.

The values below define when the specified targets are to be *set* as being ready for deletion, not *when* they are actually deleted. The actual deletion occurs only as per the cron job scheduling; or when PrintShop Mail Connect is started (if **Run at application start up** is selected); or when the **Run Now** button is pressed.

- **Enable clean-up service**: Check to enable the Clean-up services. When checked, either or both of the *Database clean-up* and *File clean-up* services can be set individually.
  If the box is not checked, then no Clean-up will occur.
- **Run at application start up**: Click to start the clean-up service when the Designer module is opened.
- **Run according to the cron schedule**: Enter the interval at which the Clean-up service runs.
  To understand how to write a cron job schedule, please refer to the Quartz Scheduler tutorial: http://www.quartz-scheduler.org/documentation/quartz-2.3.0/tutorials/tutorial-lesson-06.html.

  > **Note**
  >
  > If the **Product managing the service** is set to Designer, then the Designer *must be running* at the time that the cron job is scheduled, for the Clean-up to run.

- **Product managing the service**: Select which of the applications will run the service.For PrintShop Mail Connect the Designer is the only choice.
- **Database Clean-up Service**:
  - **Allow database clean-up service**: Select this checkbox to enable the database Clean-up settings, and enable the actual clean-up.

- **Threads to use for database deletions**: The number of Threads to be used in the clean-up. PrintShop Mail Connect is a multi-threaded application, and the clean-up is likewise.

  > **Tip**
  >
  > The default number of threads is considered the best compromise for running both clean-up and production jobs simultaneously. If experience suggests that the clean-up is not running efficiently, then upping the number of threads here would be recommended. Conversely, if production appears to be suffering courtesy of the clean-up process, then reduce the number of threads here.
  >
  > In general, higher end machines (those with multiple cores) will allow a higher numbers of threads, whilst low end machines will perform better with a lower number of threads.

- **Number of entities in each deletion batch**: The number of entities to be deleted at a time. This is done to break the clean-up into smaller chunks. This improves PrintShop Mail Connect clean-up responsiveness, whilst the clean-up is occurring.
  The number selected here applies to all the following settings.
  i.e. a selection of 1,000 would delete 1,000 data records within a **Data Set**, 1,000 content items within a **Content Set**, and so on.
- **Minimum time to retain Data Sets**: The minimum time a Data Set (and all the records it contains) is retained within the database before being set for deletion.

  > **Tip**
  >
  > In order to prevent attempts at deleting database objects which might still be in use, it is recommended that all **Minimum time** retention values *should always be set to at least the length of your longest job*. Preferably with some extra time added, for good measure.

> For example, if your longest job takes 45 to 50 minutes to run, then set the retention time to 55 minutes (or 1 hour) to ensure that all the database objects created during job processing survive for the duration of the job.

    - **Minimum time to retain Content Sets**: The minimum time a Content Set (and all the content items it contains) is retained within the database before being set for deletion.
    - **Minimum time to retain Job Sets**: The minimum time a Job Set (and all the jobs information it contains) is retained within the database before being set for deletion.
    - **Minimum time to retain Managed Files**: The minimum time file references (to files such as data mapping configurations and templates) are retained within the database before being set for deletion.
    - **Minimum time to retain other entities**: The minimum time any orphaned data (such as Finishing tables, Media tables, DataModels and Properties tables) are retained within the database before being set for deletion.
- **Database Partition Settings**:
    - **Use Database Partitioning**: Select to use Database Partitioning.
        - **Empty partition count**: The number of empty partitions that are created each clean-up run. This defaults to 24.
        - **Partition Size**: Enter the length of time before partitions are switched. This can be entered in minutes, hours, days, weeks or months.
- **File Clean-up Service**:
    - **Allow file clean-up service**: Check to automatically detect orphan files and set them for deletion. Orphan files could be resources and internal files used by Connect, but which are not needed by any running job.
        - **Minimum time to retain orphaned files**: The minimum time during which orphaned files are kept in the database before being set for deletion.

The Clean-up Services preferences also provides you with buttons to :

- **Run Now**: This will run the clean-up service immediately.
- **Restore Defaults**. This option restores the preferences to Defaults. This applies to the current Preferences page only, but not other Preferences.
- **Apply**: This option applies the settings made within the current Preferences page, but does not close the Preferences dialog.

Note that this setting only impacts how dates are displayed in the Data Model. It does not change how dates are stored in the Connect database, nor how they are displayed in a template.

Database Connection preferences

Dialog used to change the PrintShop Mail Connect back-end Database.
This dialog supports the swapping of the back-end database between various vendor databases. Note, however, that the alternate vendor database(s) *must already be installed and available* in order to swap to them.

This is not a migration tool. It is a simple connection tool, that enables shifting to a different back-end database. Any existing data will **not** be transferred/ migrated between the databases, and any existing File Stores will be cleansed by the Clean-up Service after the swap.

When shifting to a different back-end database, the changes won't be applied until PrintShop Mail Connect is restarted. Including the Connect services. A full machine restart is recommended, as this provides the cleanest restart of all the services.

- **Basic Connection Settings** selections:
  - **Database vendor**: Select the database type from the drop down list.

    > **Note**
    >
    > Moving from one vendor database to another will reset all screen selections to defaults, regardless of what may have been previously selected.

  - **Database URL**: This is a read-only summation of the current database connection settings.

> **Tip**
>
> If the **Test Connection** button shows that the database cannot be successfully connected to using the selected settings, then the contents of this field could be used to try to connect to the database outside of PrintShop Mail Connect. This should help in determining and refining the acceptable connection options.

- **Hostname**: Enter the IP Address or alias of the server where database resides.
- **Database Instance Name**: Enter an existing Microsoft SQL Server's instance name.
  This option only applies to existing Microsoft SQL Server instances, and not for MariaDB or MySQL.
- **Port**: Enter Port number. The defaults are those which the vendors use by default.
- **Schema**: The individual database schema, within the vendor database.

> **Note**
>
> If a previously non-existent schema were chosen here, then a new schema of that name will be created within the database when the back-end database swap is applied. The tables within that schema, though, will not be created until Connect is restarted.

- **Username**: Enter the database login username.

> **Tip**
>
> It is considered best practice for this user to have root privileges.

- **Password**: Enter the password associated with selected username.

- **Advanced Connection Settings** selections:
  - **Maximum concurrent threads**: This option sets the maximum database threads. The maximum setting is determined by the specific capabilities of the machine

Connect is installed upon (CPU speed and the amount of cores being the major determinants).

> **Tip**
>
> Leaving this value set to the default maximum *should* be the best option in most circumstances.
> We recommended this entry be left at the default value.

- **Custom database parameters** table: These are extra parameters which are appended to the database connection URL. The default values are those which have been determined to be useful in connecting to specific vendor databases. For example, in order to use Integrated Authentication with Windows, specify the property `integratedAuthentication` with the value `true`, which adds `integratedAuthentication=true` to the URL.
  - **Property**: These are free field text fields.

    > **Note**
    >
    > These fields and their associated values get appended to the JDBC connection and therefore must follow all rules regarding acceptable URL addresses for such.

  - **Value**: The value applied to the associated Property.
  - **Add**: Used to add extra Property values to the table.
  - **Delete**: Used to remove existing Property values from the table.
- **Test Connection**: Use to test if current connection settings will connect to the specified database.
- **Restore Defaults**: Will restore the settings to PrintShop Mail Connect HyperSQL standard defaults.
- **Apply**: When a database connection is confirmed as correct this button becomes active, and is used to actually apply the database swap.

The Database Connection preferences also provides you with buttons to :

- **Test Connection**: This will run a test on the current Database Connection settings.
- **Restore Defaults**. This option restores the preferences to Defaults. This applies to the current Preferences page only, but not other Preferences.
- **Apply**: This option applies the settings made within the current Preferences page, but does not close the Preferences dialog.

Editing preferences

These preferences define different editing options in the Designer module.

- **Object Resizing for elements**: This defines in which contexts to enable the resizing of elements (including Positioned and Inline boxes). Resizing elements may cause layouts to produce undesirable results especially when using Foundation templates.
  - **Enable for Print Context**: Check to enable resizing in the Print contexts.
  - **Enable for Email Context**: Check to enable resizing in the Email contexts.
- **Detail tables preview limit**
  - **Maximum number of records to show in preview**: This setting limits the number of records to show in detail tables in the Designer (Preview tab). The lower this number, the shorter the time before the preview or the next record will appear.
- **Dragging data fields to the editor**. When you drag-and-drop a data field into a template, a placeholder appears in the template and a script is added (see "Inserting variable data directly (drag-and-drop)" on page 351). This option specifies what to use as the script's selector.
  - **New scripts use a class selector (recommended)**: The placeholder is wrapped in a Span element, and the Span's class is used as selector. This is recommended because it is faster (see: "Optimizing scripts" on page 408).
  - **New scripts perform a search and replace**: The placeholder text is used as selector.

> **Tip**
>
> Hold the **ALT** key while dragging to change this behavior on the fly.

CSS options

A CSS preprocessor is a CSS extension language that allows you to enhance CSS with code (variables, for example) and then compile it into plain CSS. CSS Preprocessor **Sass** is integrated in Connect.
For more information about Sass, see: https://sass-lang.com/.

The CSS options relate to the way Sass (.scss) files are compiled in Connect. See also: "Using a Sass file" on page 307 in the Online Help: https://help.objectiflune.com/en/planetpress-connect-user-guide/2022.1/.

- **Output style**: This setting determines the use of spaces and indentation in the compiled CSS file. For examples of the available output styles, see: https://sass-lang.com/documentation/file.SASS_REFERENCE.html#output_style.
  - **Compact**: Each CSS rule takes up only one line, with every property defined on that line. Nested rules are placed next to each other with no new line, while separate groups of rules have new lines between them.
  - **Compressed**: This output style minifies the output. It has no whitespace except that necessary to separate selectors and a new line at the end of the file. It also includes some other minor compressions, such as choosing the smallest representation for colors. It's not meant to be human-readable.
  - **Expanded**: This is the default output style. Each property and rule take up one line. Properties are indented within the rules, but the rules aren't indented in any special way.
  - **Nested**: Each property has its own line, but the indentation isn't constant. Each rule is indented based on how deeply an element is nested in the HTML and CSS structure.
- **Auto compile on saving .scss files**: When this option is checked, a .scss file is compiled into a .css file whenever you save it, overwriting any previously compiled version of the .css file. Also, when a *partial* .scss file is saved all .scss files are recompiled. (A partial .scss file is meant to be imported in another .scss file. Its name starts with an underscore.)
  By default this option is enabled.

Note

- Re-compiling a .scss file overwrites any manual changes made to the .css file.
- Partial .scss files cannot be compiled.
- Single line comments (//...) are not added to the compiled .css file, whereas multi-line comments (/* ... */) are maintained. Multi-line comments should be added within the brackets of the definition they apply to, to make sure they appear in the correct place in the compiled css.

Color options

Many of the colors in the user interface of Connect Designer can be adjusted. Click the small colored square next to the field that holds the default color value, to open the Color dialog and pick a color (see "Color Picker" on page 492 in the Online Help: https://help.objectiflune.com/en/planetpress-connect-user-guide/2022.1/).

- **Edges**: The edges around elements in a section in the Workspace. Click the Show Edges icon to toggle the visibility of these edges.
  - **Box Objects**: This color highlights positioned boxes, inline boxes and Div elements; see "Boxes" on page 256.
  - **Table**: This color highlights tables, and the rows and columns in tables; see "Table" on page 278.
  - **Resizable Table**: This color highlights tables for which the option Allow resizing has been checked when adding the table; see "Table" on page 278.
  - **Forms**: This color highlights forms; see Forms.
  - **Shared Content**: This color highlights shared content, such as shared snippets; see "Snippets" on page 285.
- **Margin and guides**: These settings only apply to Print sections.
  - **Guides**: This is the color for rulers that can help position content correctly; see "Guides" on page 314.
  - **Margins**: This color delineates the content area on a page; see "Pages" on page 147.
  - **Bleed box**: This color delineates the printable area on a page; see "Page settings: size, margins and bleed" on page 148.

- **Master pages**: These edges are only visible on Master pages; see "Master Pages" on page 156.
  - **Header and Footer Margin**: This color highlights the header and footer margin set for the Master page; see "Adding a header and footer" on page 157.
  - **Objects**: This color highlights all elements on the Master page.
- **Script Result Highlighter**:
  - **Results**: Hovering over a script in the Scripts pane highlights content that will be affected by the script; see "Personalizing content" on page 338.

Images preferences

- **Transparent PDF image preview**: Check this option so that PDF resources added to the template (including in the Master Page and Media) display using transparency. Note that this can affect display performance (showing transparent PDFs is slower) but will not affect output speed.

The Editing preferences also provides you with buttons to :

- **Restore Defaults**. This option restores the preferences to Defaults. This applies to the current Preferences page only, but not other Preferences.
- **Apply**: This option applies the settings made within the current Preferences page, but does not close the Preferences dialog.

Email preferences

Email (General) preferences

- **Default From Group**:
  - **Name**: Enter the name that is set by default in the "From name" field in the Send Email and Send Test Email dialogs ("Send (Test) Email" on page 543).
  - **Email Address**: Enter the email that is set by default in the "From Email" field in the Send Email and Send Test Email dialogs ("Send (Test) Email" on page 543).
- **Litmus account Group**:
  - **Email Test address**: If you have a Litmus account, enter the test address to use when sending a test email (see "Send (Test) Email" on page 543 in the Online Help: https://help.objectiflune.com/en/planetpress-connect-user-guide/2022.1/). For more information on Litmus, please see http://litmus.com/.

Email (SMTP) preferences

SMTP server presets can be selected when sending emails using either the Send Email or Send Test Email dialog. (See "Send (Test) Email" on page 543 and "Email header settings" on page 180 in the Online Help: https://help.objectiflune.com/en/planetpress-connect-user-guide/2022.1/). For all presets, the password is not saved and must be re-entered when sending emails.

- The **Add**, **Edit** and **Delete** buttons let you create and manage the presets.
- **SMTP Host Settings**: These settings can be made or edited after clicking the Add or Edit button.
  - **Name**: The name of the preset. This will show up in the Send Email dialog.
  - **Host**: The SMTP server through which the emails are to be sent. Can be a host (mail.domain.com) or an IP address.

    > **Tip**
    >
    > Gmail only allows Connect to be used as an SMTP client when "Access for less secure apps" is enabled in the Google account settings.

  - **Port**: The specified port number will be added to the host name, for example: smtp.mandrillapp.com:465.

    > **Note**
    >
    > If the mail server supports it, the connection will be encrypted without the need to send the server a STARTTLS instruction when port 465 is used.

  - **Use authentication**: Check if a user name and password are needed to send emails through the host.
  - **Send STARTTLS**: Enabled if authentication is checked. With STARTTLS the client negotiates with the mail server to use some form of encryption, usually a version of Transport Layer Security (TLS). Since this improves security it is recommend to enable this option if you use port 25 (the default port), 2525, or 587.
    Note that the email will not be sent if the SMTP server does not support TLS or SSL (an older encryption type).

This option is ignored when port 465 is used.

- **User**: Enter the user name used to connect to the SMTP server.

- **Restore Defaults**: There are three default presets, each for working with a different Email Service Provider (ESP): Mandrilapp.com, Sendgrid and Mailgun (see "Using an ESP with PrintShop Mail Connect" on page 1021).
- **Apply**: Apply the new settings without closing the Preferences dialog.

The Email preferences also provides you with buttons to :

- **Restore Defaults**. This option restores the preferences to Defaults. This applies to the current Preferences page only, but not other Preferences.
- **Apply**: This option applies the settings made within the current Preferences page, but does not close the Preferences dialog.

Emmet preferences

Emmet is a framework that enables the lightning-fast creation of HTML code though the use of a simple and effective shortcut language resembling CSS Selectors (see Emmet Abbreviations: "Emmet" on page 169. The Emmet functionality is available in the HTML and CSS source editors of Connect Designer. Emmet transforms abbreviations for HTML elements and CSS properties to the respective source code.

This is, for example, the abbreviation for a element with the class row:

```
div.row
```

On pressing the Tab key, this abbreviation is transformed to:

```
<div class="row"></div>
```

To learn more about Emmet itself, please see their website Emmet.io and their documentation: http://docs.emmet.io/. Emmet is a plugin. All options listed below are Emmet's default options. They are not specifically adjusted for Connect.

Common Emmet preferences

- **Expand abbreviations by Tab key**: Check to enable the Expand Abbreviation function.
- **... in files with extension**: Enter a comma-separated list of all file extensions in which expand abbreviation will work.

- **Upgrade web editors**: This Emmet option doesn't affect how Emmet works in Connect Designer.
- **Extensions Path**: Choose a folder where to put json and js files to extend Emmet. This includes custom snippets, preferences and syntax profiles. For more information see http://docs.emmet.io/customization/.

Emmet Abbreviation preferences

This Preferences tab lets you add and manage custom abbreviations. All standard abbreviations can be found in Emmet's documentation: http://docs.emmet.io/abbreviations/syntax/.

If there is no need to transform the text while expanding it, create an Emmet snippet instead (see below).

- **New**: Add a new abbreviation.
  - **Name**: The name of the abbreviation is also its trigger.
  - **Context**: The context in which the abbreviation is enabled (HTML, CSS, etc.).
  - **Description**: A short description of the abbreviation .
  - **Pattern**: This defines what an abbreviation expands to. Since Emmet is mostly used for writing HTML/XML tags, abbreviation definition uses XML format to describe elements; see http://docs.emmet.io/abbreviations/types/.
  - **Automatically insert**: This standard option doesn't affect how Emmet works in Connect Designer.
- **Edit**: Edit the currently selected abbreviation.
- **Remove**: Remove the currently selected abbreviation.
- **Import**: Click to open a browse dialog to import an XML file containing exported abbreviations. The imported abbreviations are added to the current list.
- **Export**: Click to open a Save as dialog to export all the abbreviations in an XML file that can be shared and re-imported.
- **Preview box**: Shows what the selected abbreviation is expanded to.
- **Restore Defaults**: clear all custom abbreviations.
- To temporarily disable an abbreviation, uncheck the checkbox next to the name of the abbreviation in the list.

Emmet Output preferences

The Output Preferences dialog is used to control how the expanded (output) code behaves when expanding abbreviations and snippets. There are 6 different dialogs to control output and, while they all have identical options, they control different output types: CSS, HAML, HTML, XML, XSL and the "Default" one controlling the rest of the types.

These options are equivalent to Emmet's syntaxProfiles.json feature (http://docs.emmet.io/customization/syntax-profiles/).

Emmet Snippets preferences

Emmet Snippet are similar to abbreviations in that they are expanded when the Tab key is pressed, but they are just blocks of plain text. Anything in a snippet will be outputted “as is”, without any transformation.

- **New**: Click to create a new snippet.
  - **Name**: The name of the abbreviation is also its trigger.
  - **Context**: The context in which the snippet is enabled (HTML, CSS, etc.).
  - **Description**: A short description of the snippet.
  - **Pattern**: The pattern defines what a snippet expands to.
  - **Automatically insert**: This option doesn't affect how Emmet works in Connect Designer.
- **Edit**: Modify the currently selected snippet.
- **Remove**: Remove the currently selected snippet from the list.
- **Import**: Click to open a browse dialog to import an XML file containing exported snippets. The imported snippets are added to the current list.
- **Export**: Click to open a Save as dialog to export all the snippets in an XML file that can be shared and re-imported.
- **Preview box**: Shows what the selected snippet is expanded to.
- To temporarily disable a snippet, uncheck the checkbox next to the name of the snippet in the list.

Emmet Variables preferences

Variables are placeholders used in Emmet snippets to output predefined data. For example, the html:5 snippet of HTML syntax has the following definition:

```
<!doctype html>\n<html lang="${lang}">...</body>\n</html>
```

In the example above, ${lang} is used to refer lang variable defined in variables below. If your primary language is, for example, Russian, you can simply override lang variable with ru value and keep the original snippets. Also, you can override variable values with inline abbreviation attributes: html:5[lang=ru].

- **Name**: The name of the variable. This should be a single alphanumeric string with no spaces or special characters. For example, the myVar name is referred to as ${myVar}.
- **Value**: The value of the variable when the snippet is expanded.
- **New**: Click to create a new variable and define its name and value.
- **Edit**: Click to modify the currently selected Variable.
- **Remove**: Click to delete the currently selected Variable.

The Emmet preferences also provides you with buttons to :

- **Reload Engine**. This allows you to reload the Emmet engine based upon the current settings.
- **Restore Defaults**. This option restores the preferences to Defaults. This applies to the current Preferences page only, but not other Preferences.
- **Apply**: This option applies the settings made within the current Preferences page, but does not close the Preferences dialog.

Engines preferences

See Engines Preferences.

Hardware for Digital Signing preferences

Connect can access secure hardware devices (USB Tokens, Smart Cards, and Hardware Security Modules) through PKCS#11, for use in creating PDF digital signatures.

This Preferences page allows for the configuration of these secure hardware devices. It caters for the two conflicting needs of Connect users.

1. Connect user wanting to create digital signatures using a secure device have to configure the DLL for using the secure device, so Connect can find the DLL when a digital signature is necessary.
2. The Connect administrator wanting the configuration of DLLs loaded by Connect Server, Designer, and their engines to happen outside of Presets, Templates, or other resources.

This so the administrator can control what DLLs are loaded instead of allowing user created configurations triggering third party DLLs at runtime.

This preferences page caters for both needs.

The **PKCS#11 Modules and Tokens** table contains a list of Hardware devices that can be used by Connect.
Hardware devices can be added, modified or deleted from the table using the following buttons:

- **Add**: Add a new hardware device to the list. This launches the Add PKCS#11 Module dialog.
- **Edit**: Edit that information relating to an existing hardware device. This launches the Edit PKCS#11 Module dialog.
- **Delete**: Remove an existing hardware device from the list.

The Hardware for Digital Signing preferences also provides you with buttons to :

- **Restore Defaults**. This option restores the preferences to Defaults. This applies to the current Preferences page only, but not other Preferences.
- **Apply**: This option applies the settings made within the current Preferences page, but does not close the Preferences dialog.

Language preferences

- **Display language**: Select a language from the drop-down list to be used as the language of the User Interface (after the software is restarted).
- **Default Locale**: The default locale sets the locale for new templates. By default this is the system's locale. The locale can be changed per template; see "Locale" on page 336.
  - Select System Locale to use the operating system's locale settings.
  - Select Explicit Locale to choose a static locale from the drop-down list.

Default Locale

The Default Locale preferences are only available in the Designer Preferences. This setting determines the locale for new templates. By default this is the system's locale. Select *System Locale* to use the operating system's locale settings. Select *Explicit Locale* to choose a specific

locale from the drop-down **Explicit Locale** list. The Locale can be changed on a per template basis. See "Locale" on page 336.

The Language Settings and Default Locale preferences also provides you with buttons to :

- **Restore Defaults**. This option restores the preferences to Defaults. This applies to the current Preferences page only, but not other Preferences.
- **Apply**: This option applies the settings made within the current Preferences page, but does not close the Preferences dialog.

Logging preferences

PrintShop Mail Connect logs the activities it undertakes whilst running. New Connect logs are created daily and are held for a period before they are automatically deleted.

The settings on this page determine the level of logging and how long the log files should be retained.

These log files can be an essential resource when diagnosing issues with OL Support. The logging settings are global to ***all*** Connect applications and the Logging Preferences can be adjusted from within any of the Designer and Print Manager dialogs.

The settings are:

- **Overall Logging Level** selection: Select the overall Connect Logging level. This controls how much logging Connect will do.
  By default the logging level is set at the midpoint of *Info*, but it can be set higher to include more logging (*All*, *Trace*, *Debug*), or lower to reduce the amount of logging (*Warning*, *Error*).
  The logging is hierarchical, with each Logging Level selection containing all of the Logging Levels below it. The default logging *Info* Level contains all *Error* and *Warning* entries plus informational log entries.

Warning

Higher logging settings will have an impact upon Connect production speeds, as well as leading to substantially larger log files.

We recommended leaving the logging level to *Info* and only using the higher levels of

ogging in conjunction with advice from OL support.

- **Rollover policy** selection: Chose whether to retain Connect log files for a certain number of days (**Daily logs**) or based upon some predetermined hard disk usage limitations (**Size-based logs**). The selection are as follows:
  - **Daily Logs**: Use this setting to determine how many days Connect Logs are to be kept, before they are deleted.
    - **Number of days to retain logs**: This value only impacts upon historic (closed) logs.
      Chose between 1 and 99,999 days.
      The default value is set to 10 days for a new installation and 99,999 days on existing installations (to preserve backward compatibility).
  - **Size-based logs**: Use this setting to restrict log file size, and to keep only a specified number of them.
    By combining the maximize individual log file size with the amount of log files to retain, this effectively allows a hard disk space usage limitation to be placed upon the logging process.
    - **Maximum size for log file**: This sets the maximum size a log file can reach before the logging system creates a new file.
    - **Number of files to keep**:This sets the maximum number of log files kept in the log folder. The default value is set to 50 for a new Connect installation and 99, 999 for an existing installation (to preserve backward compatibility).
- **Logging pattern** edit box: This edit box determines the formatting of the individual log entries.
  By default it is set to a date/time value which allows for simple searching within log files.

> **Warning**
>
> We recommend leaving the Logging pattern to the default value.
>
> If you do need to change the Logging pattern, please see the Pattern Formatting guide (https://logback.qos.ch/manual/layouts.html#conversionWord) for help in doing so.

- **Log message preview** display: This displays a real time example of the format and content of individual log file entries, based upon the **Logging pattern** setting.

> **Note**
>
> The **language** of the log files is determined by the Language preferences.

Advanced settings

The Advanced settings over-ride the Overall logging settings, and provide a greater level of logging granularity.

> **Warning**
>
> Higher logging settings will have an impact upon Connect production speeds, as well as leading to substantially larger log files.
>
> The Advanced Log Settings should only be set in conjunction with advice from OL support, to ensure that only the most relevant settings are set to the higher logging levels.

This Preferences page allows you to add ( ) or remove ( ) individual Connect Packages, or change their logging settings ( ).

The Log Setting and Advanced Log Setting preferences also provides you with buttons to :

- **Restore Defaults**. This option restores the preferences to Defaults. This applies to the current Preferences page only, but not other Preferences.
- **Apply**: This option applies the settings made within the current Preferences page, but does not close the Preferences dialog.

Parallel Processing preferences

See "Parallel Processing preferences" on page 88.

Print preferences

Available Printers preferences

The Available Printers preferences control which printer definitions are available when generating print output or creating Output Presets. Any printer that is unchecked in this dialog will not be visible in the "Model" drop-down of the Print Options dialog; see "Print options" on page 704 and "Adding print output Models to the Print Wizard" on page 1005.

Available Printer Preferences:

- **Selected Printers**: Lists the available Printer Definition Files in the system. Note that these are not installed Windows printers or printer queues, but PrintShop Mail Connect Printer Definition Files.
- **Printer checkbox**: This checkbox selects/deselects all printers in the list. Click to check all, click again to uncheck all.

General Print preferences

In order to print, the Designer connects to a Connect Server. To enable communication between the Designer and the Connect Server, settings have to be made under "Preferences" on page 370.

Print Measurements preferences

- **Units**: Use the drop-down to specify the default measurements system used for dimensions of the template and boxes. In addition it defines the coordinates/position of box elements.
  The default unit will be added automatically when geometry values are entered without a unit in the Attributes pane or in the Box Properties dialog.
- **Flip insert guide axis**: Check this option to flip the axis on which guides are inserted. Normally, dragging a guide from a horizontal ruler inserts a horizontal guide (see "Guides" on page 314). With this option checked, dragging a guide from a horizontal ruler inserts a vertical guide.

The Print preferences also provides you with buttons to :

- **Test Print Server URL**. This button is only available for the General Print Preferences. It tests the Print Server URL settings made within that Preferences page.

- **Restore Defaults**. This option restores the preferences to Defaults. This applies to the current Preferences page only, but not other Preferences.
- **Apply**: This option applies the settings made within the current Preferences page, but does not close the Preferences dialog.

Sample Projects preferences

Sample Project deployment settings

- **Workflow Configurations**
  - **Encoding**: By default, Sample Projects set the encoding of the Workflow configuration that they create to the encoding of the system on which the Designer runs (**system default**).
    If Workflow runs on another machine, and that system has a different active code page, Workflow may be unable to correctly interpret all the characters that are used in the configuration file (in path or file names, for example). To prevent this, you may select an encoding from the list.

Save preferences

The saving preferences are a way control if and how often PrintShop Mail Connect saves your work in the background, and if how many backup files it creates when you save the template or data mapping configuration. See also: "Saving a template" on page 110.

Auto Save

After a template or data mapping configuration has been saved for the first time, Connect Designer can auto save it with a regular interval.

- **Enable**: activate the Auto Save function.
- **Interval (minutes)**: enter a number of minutes, e.g. 3 to auto-save the template or data mapping configuration every 3 minutes.

Auto Backup

Connect Designer can automatically create a backup file when you **manually** save a template or data mapping configuration. The Auto Save function does **not** cause backup files to be created.

- **Enable**: activate the Auto Backup function.
- **Revisions to keep**: Enter the maximum number of backup files. When the maximum is reached, Auto Backup will overwrite the oldest file.
- **Destination**: Select the directory in which the backups should be stored.
  - **Original**: the directory in which the original file is stored.
  - **Other directory**: use the **Browse** button to select another directory.

Backup files have the same name as the original file with two underscores and a progressive number (without leading zeros) at the end: **originalname__1.OL-template**, **originalname__2.OL-template**, etc.

The Save preferences also provides you with buttons to :

- **Restore Defaults**. This option restores the preferences to Defaults. This applies to the current Preferences page only, but not other Preferences.
- **Apply**: This option applies the settings made within the current Preferences page, but does not close the Preferences dialog.

Scripting preferences

The Scripting preferences define different options related to scripting within PrintShop Mail Connect. See also: "Testing scripts" on page 403.

- **General**:
  - **Script timeout at design time (sec)**: In Preview mode or when running the Script Profiler (see the Profile Scripts dialog), a long running script is stopped after the amount of time set here. The default is 2 seconds, the minimum is 1 second.
  - **Expanded script quotes style**: When the Expand button in a Script Wizard is clicked, the expanded script will use either **double (")** or **single (')** quotation marks.

    > **Tip**
    >
    > Using single quotation marks in a script simplifies adding HTML fragments, which typically use double quotes.

- **Designer scripting profiling group**:

- **Number of iterations**: Enter the number of times to run scripts when running the Profile Scripts dialog. The default is 1000. Accepted values are 1 to 1000000000. Yes, that's 1 billion - which would take a *long* time to run!

The Scripting preferences also provides you with buttons to :

- **Restore Defaults**. This option restores the preferences to Defaults. This applies to the current Preferences page only, but not other Preferences.
- **Apply**: This option applies the settings made within the current Preferences page, but does not close the Preferences dialog.

Writing your own scripts

Personalization can be taken a lot further than just inserting names and addresses, and hiding or showing text or images. Every bit of information in your communications can be made entirely personal, using scripts.

A script is a small set of instructions to the program, written in JavaScript.
When Connect generates the actual output – letters or emails -, it opens a record set and merges it with the template. It takes each record, one by one, and runs all scripts for it (in a specific order, see "The script flow: when scripts run" on page 413).

Any kind of personalization is done via scripts, but you don't have to write every script yourself. In many situations you can use a **Script Wizard**, which will do that for you (see "Personalizing content" on page 338).
For a block of variable data, such as an address, the **Text Script Wizard** is a perfect fit.
Paragraphs can be made conditional with a **Conditional Script Wizard**.
For dynamic images, you can use the **Dynamic Image Script Wizard**.
In an Email context, you are provided with a number of **Email Script Wizards** to set the sender, the recipients and the subject of the email.

However, when you want to do something that goes beyond what you can do with a Wizard, like creating a conditional paragraph with a condition that is based on a combination of data fields, you have to write the script yourself.

This topic explains how scripts work and how you can create and write a script yourself.

Script types

There are three types of scripts in the Designer: **Control Scripts**, **Standard Scripts** and **Post Pagination Scripts**.

Control Scripts

When output is generated from a template, Control Scripts run **before** all other scripts, when a record is merged with a context. They determine how different sections of the context are handled. They can, for example, make the page numbering continue over all Print sections, split Email attachments, or omit Print sections from the output.

Some knowledge of JavaScript is needed to edit Control Scripts, just as for any other self-made scripts; see "Writing your own scripts" on the previous page. Some functionality is provided as a Scripting Wizard, for example Conditional Print Sections.

Control Scripts don't touch the content of the sections themselves, but they change the way a template is outputted. For more information about Control Scripts and their use, see "Control Scripts" on page 429.

Standard Scripts

Standard Scripts can change the contents of sections in a template. This type of script must have a **selector**. The selector can be text, an HTML element and/or a CSS selector (see "Selectors in Connect" on page 414). Running a Standard Script starts with looking for pieces of content in the template that match the script's selector.
The results of this query can vary from one occurrence of a simple text (for example: @EMAIL@) to a large collection of HTML elements. For example, when the selector is **p**, the HTML tag for a paragraph, all paragraphs will be collected and passed to the script.

> **Tip**
>
> Hover over the name of a script in the **Scripts** pane to highlight parts of the template that are affected by the script.

Next, the script can modify the selected pieces of content, using values from the record that is merged to the template at the time the script runs. It can, for example, hide, replace or add text or change the style of those pieces of content. This is how scripts personalize documents.

Note

In a Print context, the scripts in the Scripts pane run once for each section and then once for each Master Page (see "Master Pages" on page 156).

Tip

Content added by a script isn't visible in Design mode, but is visible and can be inspected in Preview mode.

Post Pagination Scripts

Post Pagination Scripts are run in a **Print** context **after** the content has been paginated. Because they can search through the output of all Print sections, and modify Print sections (one at a time), they may be used to create a Table Of Contents (TOC), as explained in the topic: "Creating a Table Of Contents" on page 447.

For more information see "Post Pagination Scripts" on page 445.

Creating a new Standard Script

Writing a Standard Script starts with this procedure.

1. On the **Scripts** pane at the bottom left, click **New**. A new script appears in the list. Double-click on it to open it.
2. Change the name of the script, so that it reflects what the script does.

   Note

   Scripts can only have the same name when they are not in the same folder. (See "Managing scripts" on page 400.)

3. Choose which kind of **selector** you want to use. Running a Standard Script starts with searching the template for pieces of content that match the script's selector. The collected pieces of content are passed on to the script (all at the same time, or one by one - see

"Setting the scope of a script" on page 399), so that the script can modify them.
The selector can be:

- **Text**, for example: @lastname@, or {sender}. The text doesn't have to have any special characters, but special characters do make it easier to recognize the text for yourself. In the Script Wizard, click **Text** and type the text to find.
- A **selector** (HTML/CSS):
  - HTML elements of a certain **type**, such as a paragraph: <p>. In the Script Wizard, click **Selector** and type the HTML tag in the Selector field without the angle brackets: p.
  - HTML elements with a specific CSS **class** (eg. green). In the Script Wizard, click **Selector** and type the class name in the Selector field, preceded by a dot: .green.
  - An HTML element with a specific **ID** (eg. intro). In the Script Wizard, click **Selector** and type the ID in the Selector field, preceded by #: #intro.
    In an HTML file, each ID should be unique. This means that a particular ID can be used only once in each section.
  - Etcetera. See https://www.w3schools.com/cssref/css_selectors.asp for more selectors and combinations of selectors; also see "Selectors in Connect" on page 414 for selectors that can only be used in Connect.
- A **selector and text**. This is text inside an HTML element (or several HTML elements) with a specific HTML tag, CSS class or ID. In the Script Wizard, click **Selector and Text**.

Tip

When output speed matters, choose **selector** or **selector and text**. Searching text is a rather lengthy operation, compared to searching for HTML elements and/or CSS selectors. See also: "Optimizing scripts" on page 408.

Quick-start a script for a specific ID or class

There is a shorter route to create a script for an element with a specific ID or class:

1. In the template, click the element for which you want to create a script.
2. On the **Attributes** pane at the top right, type an ID or class.

3. Click the label to the left of the ID or Class input field (ID or Class) to make a new script with that ID or class as selector.

Writing a script

1. Create a new script (see: "Creating a new Standard Script" on page 395, " Adding a Control Script" on page 430 or "Adding a Post Pagination Script" on page 446), or double-click an existing script in the **Scripts** pane on the bottom left.
   If the script was made with a Script Wizard, you have to click the **Expand** button before you can start writing code. This will change the Script Wizard into an **editor window**.

   > **Warning**
   >
   > When you change an expanded text script and save it, it becomes impossible to edit the script using the Script Wizard again.

2. Write the script. Click **Apply** from time to time to see if the script works as expected. This will be visible on the **Preview** tab in the main workspace.

Syntax rules

Every script in the Designer must follow JavaScript syntax rules. For example, each statement should end with **;** and the keywords that can be used, such as **var** to declare a variable, are JavaScript keywords. There are countless tutorials available on the Internet to familiarize yourself with the JavaScript syntax.
For a simple script all that you need to know can be found on the following web pages: https://www.w3schools.com/js/js_syntax.asp and https://www.w3schools.com/js/js_if_else.asp.
A few examples can be found in a How-to: Combining record based conditions.

> **Tip**
>
> In the editor window, press **Ctrl** + **Space** to see the available features and their descriptions.
> Use the arrow keys to select a function or object and press Enter to insert it in the script.
> Type a **dot** after the name of the function or object and press Ctrl + space again to see which features are subsequently available.
> For more keyboard shortcuts, see "Keyboard shortcuts" on page 561.

Two basic code examples

Writing a script generally comes down to modifying the piece(s) of content collected from the template with the script's selector, using values, or depending on values of the record that is being merged to the template at the moment the script runs.

Modifying the template

To access and change the results of the query that is carried out with the selector (in other words: to modify the output), use the object **results**.

The following script (with the selector **p**) changes the text color of all paragraphs to red with a single line of code:

```
results.css('color', 'red')
```

It does this for each and every customer, because it does not depend on a value from the record that is being merged to the template.

Using values from the record in a script

To access the record that is being merged to the template when the script runs, use the object **record** (see "record" on page 848).

Suppose you want to display negative amounts in red and positive amounts in green. Assuming that there is an AMOUNT field in your customer data, you could write the following script (with the selector: **td.amount**, that is: table cells with the class 'amount').

```
var amount = record.fields.AMOUNT;
if (amount >= 0)
        {results.css('color', 'green');}
else if (amount < 0) {
        results.css('color', 'red');
}
```

When this script executes, it stores the value of the AMOUNT field from the current record in a variable and evaluates it. If the value is zero or higher, the color of text in the **results** - the table cells in this case - will be set to green; if the value is below zero, the text color will be set to red.

Tip

For more examples of using conditions, see this how-to: Combining record-based conditions.

If an expanded script contains errors or if there are warnings, icons appear in the overview ruler on the right hand side of the editing area. These icons are shown relative to their position in the script and do not move as you scroll down. You can click on an icon to quickly jump to the error or warning. Script errors are highlighted by a red icon, and warnings in yellow. The topmost icon will display red if any errors exist in the script at all.

Designer API

Features like **results** and **record** do not exist in the native JavaScript library. These are additional JavaScript features, designed for use in Connect scripts only. All features designed for use in the Designer are listed in the Designer's API, with a lot of examples; see "Standard Script API" on page 810.

Tip

Do you find yourself copy-pasting the same function into every new script? You can avoid this by defining your function in a Control Script. Control Scripts are executed first, so the function will then be available in all Standard Scripts and Post Pagination Scripts.

Setting the scope of a script

The selector of a script can match multiple elements. By setting the scope of the script you can determine whether you want to run the script **once**, or **once for each element** that matches the selector. The second option is especially useful when a script targets rows or cells in a Dynamic Table (see "Using scripts in Dynamic Tables" on page 426).

To set the scope of a script, click on **Options** at the bottom of the Script Editor window. The options are:

- **Result set**: The script will run once, regardless of the number of elements that match the selector. It will have access to all elements that match the selector via the `results` object (see "results" on page 972). The `each()` function could be used to iterate over

elements in the results.

- **Each matched element**: The script runs once for each element that matches the selector. The current element is accessible via the `this` object (see "this" on page 891). If the script targets (something in) a row that is bound to a detail table, the current detail record is accessible via `this.record`.

> **Note**
>
> The scope of Control Scripts can't be set, because they don't have a selector.

Managing scripts

Changing the order of execution

When a record set is merged with a template to generate output, all scripts are executed once for every record in the record set, in the order in which they appear in the **Scripts** pane at the bottom left.

The order in which scripts are executed is particularly important when one script produces content that contains a selector for another script. If the other script has already been executed, it will not run again automatically. So, scripts that produce content that contains one or more selectors for other scripts, need to come first.

To change the order in which scripts are executed:

- Click a script or a folder in the **Scripts** pane at the bottom. Drag it up or down and drop it.

> **Note**
>
> Control scripts are always executed first, regardless of where they are in the Scripts pane. They can not be excluded from execution for a specific context or section, using the execution scope of a folder; see "Execution scope" on the next page. What you can do is disable the script or the containing folder; see "Enable/disable scripts" on page 402.

Script folders

Scripts can be organized in folders. Why would you do that? For three reasons:

- Folders have an execution scope. You can specify for which contexts and sections the scripts in a folder have to run.
- Folders provide a better overview than a long unorganized list of scripts.
- Folders make it easier to change the order of execution for a bunch of scripts (see: "Changing the order of execution" on the previous page to learn why the order of execution is important). Dragging a folder up or down will cause all the scripts in that folder to be executed earlier or later, respectively.

To make a new folder on the **Scripts** pane:

1. In the **Scripts** pane, click the black triangle on the **New** button.
2. Click **Folder**. The folder will appear in the list of scripts.
3. Change the name of the new folder: right-click the folder and click **Rename**.
4. Drag scripts to the folder.

Tip

It may be helpful to put scripts that have an effect on the same context or section in one folder, because you can set the execution scope of scripts per folder (see: "Execution scope" below).

Note

Control scripts are always executed first, regardless of where they are in the Scripts pane. They can not be excluded from execution for a specific context or section, using the execution scope of a folder; see "Execution scope" below. What you can do is disable the script or the containing folder; see "Enable/disable scripts" on the facing page.

Execution scope

A particular script may be used in one context or section, but not in other contexts or sections. Nevertheless, when processing the template, the Designer tries to find the selector of each script in all contexts and sections – unless the script is located in a scripts folder for which the execution scope has been set to the relevant contexts or sections. So, setting the execution scope of a folder saves processing time.

To change the execution scope of a script:

1. Put the script in a folder; see "Script folders" on page 400.
2. Right-click the folder, and then click **Properties**.
3. Check the contexts and sections for which the scripts in this folder should run.

> **Note**
>
> Control scripts are always executed first, regardless of where they are in the Scripts pane. They can not be excluded from execution for a specific context or section, using the execution scope of a folder; see "Execution scope" on the previous page. What you can do is disable the script or the containing folder; see "Enable/disable scripts" below.

> **Tip**
>
> For more ways to optimize scripts, see "Optimizing scripts" on page 408.

Enable/disable scripts

A disabled script will not run at all when the template is merged with a record set to generate output. Disabling script execution in certain contexts or sections helps with performance, since scripts normally run, whether or not their placeholder or selector is present in your template. It is highly recommended to disable any script that is not relevant to specific sections or contexts.

When you disable a folder, all scripts in the folder will be disabled.

To enable or disable a script or a folder:

- On the **Scripts** pane, right-click the script or the folder and click **Disable** (if the script or folder was enabled) or **Enable** (if the script or folder was disabled).

> **Tip**
>
> For more ways to optimize scripts, see "Optimizing scripts" on page 408.

Import/export scripts

The easiest way to import scripts into another template is to open the template and use the **Import Resources** dialog; see "Import Resources dialog" on page 502.
Remember also to import or add any files that a script refers to.

Alternatively, scripts can be exported or imported via the Scripts pane.
To export a script:

1. On the **Scripts** pane, click on a script, and then click the **Export** button, or right-click a script and select **Export**.
2. Give the script a name and click **OK**.

To import a script:

- On the **Scripts** pane, click the **Import** button. Find the script and click **OK**

Files that a script may refer to, such as images, snippets and fonts, are not exported or imported together with a script.

Test the script to make sure that all files are present in the template and that the script's selector matches something in the content of the template; see "Testing scripts" below.

Testing scripts

The quickest way to test that scripts work as expected, is to click the **Preview** tab at the bottom of the workspace.

You can even do this while creating a new script, either with a Script Wizard or in the expanded script editor. Click **Apply** at the bottom of the script editor to see the effect of the script on the **Preview** tab of the Designer.

Note that scripts that use values of data fields can only be effective when a data file is open.
See "Loading data" on page 341.

> **Note**
>
> **Post Pagination Scripts** run only when a Print *section* is previewed or outputted.
> To verify the results of Post Pagination Scripts on a certain *Master Page*, preview the

Print section to which that Master Page is applied.

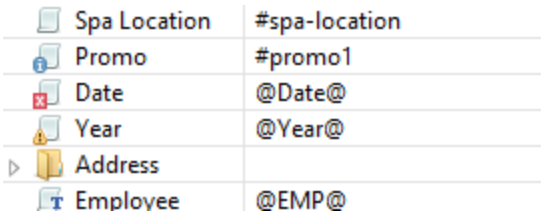
Testing for errors

Information in the Scripts pane

One way to see whether a script is functional is to take a look at the Scripts pane.

> **Tip**
>
> Hover over the name of a script in the **Scripts** pane to highlight parts of the template that are affected by the script.

Icons on the name of scripts in the **Scripts** pane can show that there is a warning, information or error. If one of these icons appears, you can hover over it to see more details.



- The information icon (i) shows that the selector of the script does not produce a result in the current section. Note that standard scripts and post pagination scripts are not executed if their selector does not match any elements. A control script is always executed, assuming the script is enabled.
- The warning icon (!) appears, for example, when a script refers to an unknown field in the record set, or when `;` is missing after a statement.
- The error icon (x) displays when the script results in an error, for example, when it uses an undeclared variable.

If an expanded script contains errors or if there are warnings, icons appear in the overview ruler on the right hand side of the editing area. These icons are shown relative to their position in the script and do not move as you scroll down. You can click on an icon to quickly jump to the error

or warning. Script errors are highlighted by a red icon, and warnings in yellow. The topmost icon will display red if any errors exist in the script at all.

> **Tip**
>
> To do a quick **search** for a certain script or piece of code, type a search text in the filter field located below the buttons in the Scripts pane.

Doing a Preflight

In addition to the icons and messages in the Scripts pane, a **preflight** can show if your scripts function as expected before generating output:

1. On the menu, select **Context > Preflight**.
2. Select **All**, or enter a selection of records. You can specify individual records separated by semi-colons (;) or ranges using dashes. For example: 2,4,6-10 would print pages 2, 4, 6, 7, 8, 9 and 10.
3. Click **OK**.

Preflight executes the template without actually producing output.
The Preflight window displays any issues once it's done. It will tell, for example, which selectors were not encountered in the template.
Double-click a script warning/error (either in the Preflight Progress dialog or in the Preflight Result view) to open the script in the script editor. The relevant line will be highlighted.

> **Tip**
>
> Be aware that scripts run in a specific order (see "The script flow: when scripts run" on page 413). When one script unintentionally influences the results of another script, changing the order of the scripts in the Scripts pane may help (see "Changing the order of execution" on page 400).

Note

An image with an unknown file extension is represented by a red cross in the output, but no error is logged unless the image refers to a local file that does not exist on disk.
Image file extensions that the software recognizes are: AFP, BMP, EPS, GIF, JPG/JPEG, PCL, PDF, PNG, PS, SVG, and TIF/TIFF.

Using the Script Debugger

The Designer's Script Debugger allows you to set breakpoints in scripts and step through them. It simulates an output run with only the current record and shows an overview of the state of all local and global variables, while the Workspace displays the partially merged document.

There are two ways to start the Script Debugger:

- Click the **Debug Scripts** button in the toolbar of the Scripts pane. The Script Debugger will pause and accept input as soon as it processes the first script.
- Right-click an enabled script in the Scripts panel and choose **Debug** from the contextual menu. The Script Debugger will pause and accept input as soon as it processes the selected script.

If the selected script is never processed, a message will pop up. This can happen when:

- The selector has no matches at any time. (This may depend on other scripts, since scripts can add or remove content.)
- The script is excluded through the properties of its parent folder.

For an explanation of the buttons in the Script Debugger, see "Script Debugger" on page 531.

Testing for speed issues

To measure the time that the execution of scripts will take:

- On the **Context** menu, click **Profile scripts**.

Profiling means running the scripts in the template, with the current record, to see how fast scripts in the **Scripts** pane execute. It helps greatly in troubleshooting performance issues caused by scripts.

After running the Script Profiler you can see in which sections the script has run:

- Hover the mouse over a value in the column **Count** to see the number of times that the script has run, per section.

You can also see the breakdown of the execution time across different execution stages:

- Hover the mouse over a value in the column **Elapsed** to see the time elapsed (in milliseconds) since the start of the session. In the Scripts Profiler, the scripts are by default sorted based on the values in the **Elapsed** column, from high to low.
- Hover the mouse over a value the column **Delta** to see the difference between the time elapsed (in milliseconds) in the previous session and in the current session.

The script execution stages are:

Query: the time it takes to find the selector in the template.

> **Tip**
>
> Looking for text is a rather lengthy operation. Use an ID or class (possibly in combination with a text) instead of a text selector to make the query faster. For more tips, see "Optimizing scripts" on the facing page.

Execution: the time it takes to execute the script. If you are an experienced JavaScript coder you may be able to optimize the code to speed up the execution of the script.

> **Tip**
>
> Functions that actually change the content of the template (for example,**append()**) are comparatively time consuming. Avoid using such functions in a loop. For more tips, see "Optimizing scripts" on the facing page.

Note that the times vary slightly per run of the Script Profiler. Run the Script Profiler a number of times and calculate an average from the results, before trying to speed up the execution of a script.

Script Profiler settings

Number of runs

By default, the Script Profiler runs on 1000 instances of all the scripts. To test on a higher or lower number of instances:

1. On the menu, select **Window > Preferences**.
2. Click **Scripting**.
3. Set a number of iterations (maximum one billion) and click OK.

Sorting

In the Scripts Profiler, the scripts are by default sorted based on the values in the **Elapsed** column, from high to low. Click any of the columns to sort the scripts according to the values in that column.

Script timeout

When testing scripts, either by toggling to Preview mode or by using the Script Profiler, a script timeout is active in the Designer, so that scripts that need a very long time to run are stopped after a set time. You can adapt this timeout to your needs, as follows:

1. On the menu, select **Window > Preferences.**
2. Click **Scripting**.
3. Set a timeout in seconds (for example: 2s) and click OK. The minimum timeout is 1 second.

> **Note**
>
> The script timeout is not active when generating output.

Optimizing scripts

In the process of output generation, the execution of scripts may take up more time than necessary. To optimize a template, it helps to disable scripts that don't have an effect on the output; see "Managing scripts" on page 400.

This topic presents a number of other ways to speed up script execution by optimizing the scripts.

Use an ID or class as selector

Scripts (except Control Scripts) start with a query. The **selector** in the second column in the **Scripts** pane is what a script looks for in the template. The selector can be a text, HTML/CSS tags, or a combination of text and HTML/CSS tags.

Looking for text in a template is a less optimized operation and may impact output speeds in longer documents. To speed up the output process, it is recommended to use an ID or class as selector instead. This narrows the scope of the search and results in a very fast query, as elements with an ID or class are indexed by Connect Designer's layout engine.

For information about how to give an element an ID or class, see "ID and class" on page 193. See also: "Quick-start a script for a specific ID or class" on page 396.

Targeting text

Text in itself cannot have an ID or class, but the element that contains it can. The smallest possible container of text is a Span. To learn how to put text inside a Span, see "Span" on page 259. Give the Span an ID or class and use that as the script's selector.

To target text in a bigger container (a paragraph, for example), change the Find method of the script to **Selector and Text**, use the ID or class of the container element as **Selector** and type the text in the **Text** field.

> **Note**
>
> When you use the **drag-and-drop** method (without pressing the ALT key) to insert variable data into a template, the script's selector is by default the **class** of the element - a Span or Div - in which the placeholder is wrapped. For more information see: "Inserting variable data directly (drag-and-drop)" on page 351.

Avoid DOM manipulations

The Scripting API of the Designer is a very powerful tool to manipulate and personalize your document. But keep in mind that DOM manipulation commands like append(), prepend(), before() and after() are resource intensive.

Try avoiding DOM modifications, especially within loops. Storing the content in a variable and appending the information after the loop is more efficient: this way, the template will be touched only once.

Example

The following example loads a snippet into a variable and uses the find() and text() commands of the Designer scripting API.

```
var labelElm = loadhtml('snippets/label.html');
for(var i = 0; i < record.tables.products.length; i++) {
       var label = labelElm.clone();
       label.find('@ProductLabel@').text(record.tables.products
[i].ProductDescription);
       results.after(label);
}
```

What's wrong with this code is that it inserts the personalized information **within** the loop. The after() command runs as many times as there are records in the detail table 'products'.

The script below is much more efficient: it adds the personalized content to a string called labelStr and only calls after() after the `for` loop.

```
var labelElm = loadhtml('snippets/label.html');
var labelStr = "";
for( var i = 0; i < record.tables.products.length); i++) {
       var label = labelElm.clone();
       label.find('@ProductLabel@').text(record.tables.products
[i].ProductDescription);
       labelStr += label;
}
results.after(labelStr);
```

Use replace()

When personalizing HTML fragments retrieved from a snippet or from the template itself, JavaScript's `replace()` method shows the best performance.

`Replace()` can only be used on Strings, while the commands `loadhtml()` and `query()` return or a QueryResult, which is a set of strings, like the `results` object.

A QueryResult allows you to perform DOM manipulations like adding and removing elements, adding and removing CSS classes etc. When the required manipulations are limited to find/replace actions, you could change the QueryResult into a string. This allows you to replace text using the `replace()` method.

For this, you could use `toString()`:

```
var labelSnippet = loadhtml('snippets/label.html').toString();
```

Or you could copy the HTML of the QueryResults to a variable:

```
var block = results.html();
```

Example

```
var labelSnippet = loadhtml('snippets/label.html').toString();
var labelStr = "";
for( var i = 0; i < record.tables.detail.length; i++) {
       var label = labelSnippet;
       label = label.replace('#', i);
       label = label.replace('@product@', record.tables.detail[i].fields
['product']);
       label = label.replace('@notes@', record.tables.detail[i].fields
['notes']);
       label = label.replace('@netweight@', record.tables.detail
[i].fields['netweight']);
       labelStr += label;
}
results.after(labelStr);
```

Tip

The `replace()` method as used in the above example replaces only the first occurrence of the search string. To **replace every occurrence** of a search string in a given string, use a **regular expression**. In the following line of code, the regular expression `/@product@/g` makes replace() search for all occurrences of the string @product@ in the `label` string:

```
label = label.replace(/@product@/g, record.tables.detail
[i].fields['product']);
```

In this example, @product@ is a pattern (to be used in a search) and g is a modifier (to find all matches rather than stopping after the first match). For more information about possible regular expressions, see https://www.w3schools.com/js/js_regexp.asp.

Replace several placeholders in one script

Suppose there are 20 different placeholders in a postcard (for the address, account and customer details, a promo code, the due date, discounts, a link to a personalized landing page etc.). Typically this would require 20 queries. Even after optimizing these scripts by using an ID as selector for those scripts, there are still 20 scripts, 20 queries to run.

If there was only one query, one single script to do all the work, the output could be generated much faster. Reducing the number of scripts improves the performance of the template. How to do this?

First, wrap the content that contains all of the placeholders in one (inline) Box and give that Box or Span an ID (on the Attributes pane). Next, create a script that uses that ID as selector. Then replace all placeholders in the script and put the content back in the template.

This is similar to working with snippets, but in this case the element is extracted from the actual template.

Example

The following script replaces all of the placeholders on a postcard. It takes advantage of the JavaScript replace() command. Assuming that the ID of the block that requires personalization is `promoblock`, the script has to have its selector set to `#promoblock`.

```
var block = results.html();
var data = record.fields;
block = block.replace('@name@',data.first + ' ' + data.last);
block = block.replace('@address@',data.address);
block = block.replace('@zip@',data.zip);
block = block.replace('@city@',data.city);
block = block.replace('@country@',data.country);
block = block.replace('@saldo@',data.saldo);
block = block.replace('@promo@',data.promo);
block = block.replace('@customercode@', data.customercode);
```

```
…
results.html(block);
```

The first line retrieves the HTML of the promo block and stores it in a variable called block. To make the code more readible, the fields from the record are stored in a variable named `data`. After replacing the placeholders by values, the script replaces the HTML of the promoblock with the personalized string.

Other resources

There are also many resources online to help learn about JavaScript performance and coding mistakes. See for example:

- JavaScript performance
- The 10 most common JavaScript mistakes
- Tips for writing efficient JavaScript.

Note that most resources on the web are about JavaScript in the *browser*, but the greatest majority of the tips do, indeed, apply to scripts in general, wherever they are used.

The script flow: when scripts run

When Connect generates the actual output – letters, web pages or emails -, it opens a record set and merges it with the template. It takes each record, one by one, and runs all scripts for it, in a specific order, as explained below.

First all Control Scripts are executed, in the order in which they appear in the Scripts pane. Control scripts don't touch the content of the sections themselves, but they change the way a template is outputted, for example by selecting or omitting sections from the output (see "Control Scripts" on page 429).

Then the Standard scripts are executed, once for each section, in the order in which they appear in the Scripts pane.
Standard scripts can change the contents of the current section in a template.
This type of script must have a **selector**: text, an HTML element and/or a CSS selector (see "Writing your own scripts" on page 393 and "Selectors in Connect" on the facing page).
Running a template script starts with looking in the current section for pieces of content that match the script's selector.
Important to note is that **if nothing matches the selector, the script is not executed.**

In a **Print context**, the Standard scripts in the Scripts pane run once for each section and then for each Master Page (see "Master Pages" on page 156). Next, each processed Master Page is put behind every page to which it should be applied.
Scripts are NOT executed again for every page.

Finally, Post Pagination Scripts run, in the order in which they appear in the Scripts pane (see "Post Pagination Scripts" on page 445).
Post Pagination Scripts are run in a **Print** context **after** the content has been paginated. Because they can search through the output of all Print sections, and modify Print sections (one at a time), they may be used to create a Table Of Contents (TOC), as explained in the topic: "Creating a Table Of Contents" on page 447.

Tip

Do you find yourself copy-pasting the same function into every new script? You can avoid this by defining your function in a Control Script. Control Scripts are executed first, so the function will then be available in all Standard Scripts and Post Pagination Scripts.

Note

Any JavaScript files included in a section run **after** the scripts in the Scripts pane.

Selectors in Connect

Selectors are patterns used to select one or more HTML elements. They were originally developed to be able to define the layout of web pages without touching their content, through Cascading Style Sheets (CSS). In Connect, since each section in a Connect template is in fact an HTML file (see "Editing HTML" on page 193), the very same selectors can be used in style sheets (see "Styling templates with CSS files" on page 296) and template scripts (see "Personalizing content" on page 338 and "Writing your own scripts" on page 393). Using selectors for scripting can increase the speed with which a template and data are merged; see "Use an ID or class as selector" on page 409.

Standard CSS selectors

Selectors are made up of one or more of the following components:

- An **HTML element**. Type the HTML tag without the angle brackets (e.g. `p`) to select all elements of that type (`p`selects all paragraphs).
- A **class**. Type the class name, preceded by a dot, e.g.: `.green`, to select HTML elements with that class.
- An **ID**. Type the ID, preceded by #, e.g.: `#intro`, to select an HTML element with that ID.
- An **attribute** of an HTML element. Type the attribute and, optionally, its value, between square brackets, e.g.: `[target]`, to select HTML elements with a matching attribute.
- A **pseudo-class**. For example, `tr:nth-child(even)` selects all even table rows.

These components can be combined in different ways. For example, p div selects all paragraphs *inside* elements, while p, div selects all paragraphs *and* all elements.

A complete list of selectors and ways to combine them, and a tool that demonstrates their use can be found at W3Schools: http://www.w3schools.com/cssref/css_selectors.asp.

A **video** about CSS and Script Selectors, can be found here: Connect with Evie 6 - CSS and Script Selectors.

Connect classes and attributes

Connect itself sometimes adds a specific class or attribute to elements in a template. Capture OnTheGo widgets, for example, have a role attribute that allows the COTG library to dictate their behaviour. Connect classes and attributes can be used in selectors, as will be explained and demonstrated below.

Connect-specific classes usually are invisible in the Designer. By opening the currently selected section in your default web browser (click the Preview HTML toolbar button) and using the browser's code or source inspector you can see most of the dynamically added classes.

> **Warning**
>
> Avoid using classes with the **__ol** prefix in your selectors. These dynamically added class names may change in future releases of the software.

Section selector

The Designer writes the name of each section to the `section` attribute of the <html> element. This attribute can be used in selectors.

Note

To make scripts run exclusively on certain sections, it is advised to put them in folders and set the execution scope of the scripts in a folder via the folder properties; see "Execution scope" on page 401.

Example

The following rule applies formatting to <h1> elements in sections of which the name starts with 'Letter':

```
[section^='Letter'] h1 {
        color: brown;
}
```

Note

To target sections as well as Master Pages, use the `body` selector without the `masterpage` or `section` selector. For example:

- Selector: `body`
- Script: `results.html('<div style="background:red;width:1in;height:2in">Hello World</div>');`

In versions prior to 2019.2 the `section` selector would work on sections as well as Master Pages. As of version 2019.2 this is no longer the case. The `section` selector now only selects sections.

Master Page selector

The Designer writes the name of each Master Page to the `masterpage` attribute of the <html> element. This attribute can be used in selectors.

Example

This script adds a box to the `body` of every Master Page.

- Selector: `[masterpage] body`
- Script: `results.html('<div style="background:red;width:1in;height:2in">Hello World</div>');`

The following script adds the box only to the Master Page called "Master Page 1".

- Selector: `[masterpage="Master page 1"] body`
- Script: `results.html('<div style="background:red;width:1in;height:2in">Hello World</div>');`

Sheet position selectors

In Print output, pages have a sheet position that depends on the options set in the "Sheet Configuration dialog" on page 546 (e.g. the Duplex and Allow Content On options). Connect gives each page - or rather, the "MediaBox" div element on that page - a class depending on their sheet position:

- .frontside
- .backside (does not apply to simplex documents)
- .contentpage
- .nocontentpage

The MediaBox contains the Master Page objects and section backgrounds. This means that these classes can only be used to format a Master Page and section background. They do not let you change the formatting of elements residing in the main text flow (e.g. a <h1> element on page 3).

Formatting Master Page objects depending on the sheet position

The following CSS rule sets the color of <h1> elements on a Master Page when that Master Page is present on the front of a sheet.

```
.frontside h1 {
        color: green;
}
```

The next style rule is a bit more specific: it colors <h1> elements on a Master Page when that Master Page is applied to the front of a sheet in Section 1:

```
[section='Section 1'] .frontside h1 {
        color: green;
}
```

The following rule hides <h1> elements on the back of a sheet on which no content (from the main text) is allowed.

```
.backside.nocontentpage .h1 {
        display: none;
}
```

Print section background selector

When you inspect a Print section in a browser, you will see that it has a element as the first child of the <body> element. Inside this there are one or more MediaBoxes: elements with the class page_mediabox. Each MediaBox contains the Media, section background and Master Page that apply to one page (see "Media" on page 159, "Master Pages" on page 156 and "Using a PDF file or other image as background" on page 142).

In the MediaBox, a Print section background is an <img> element with the ol_pdf_datamapper_input class. Its `src` attribute references the PDF file that contains the image and the `page` parameter is used to select a specific page in that PDF (as a PDF can contain more than one page).
For example:
<img src="file:/C:/Users/MyUser/Pictures/mixed.pdf?page=1" class="ol_pdf_datamapper_input">.
You can use the ol_pdf_datamapper_input class as a selector to target the section background in a style rule or script.

Placing the section background in front of the Master Page

The stacking order of elements inside each MediaBox, from bottom to top, is:

1. Media
2. Section background
3. Master Page elements

Using the .page_mediabox selector, you could change this stacking order and place the section background on top of the elements on the Master Page. Set the z-index property to a

value larger than 0 (zero) and add !important to make this style rule override the inline style declaration that normally puts the section background behind the Master Page elements:

```
.page_mediabox img.ol_pdf_datamapper_input {
        z-index: 10 !important;
}
```

Scaling the section background

The rule below downscales the section background image and keeps it in the centre of the page:

```
.page_mediabox img.ol_pdf_datamapper_input {
        transform: translate(-50%, -50%) scale(1.5, 1.5) !important;
}
```

View selectors

In the Designer, sections can be viewed on different tabs: Source, Design, Preview and - if it is a Web section - Live. In each view mode (except Source) a specific CSS class is added to the <html> element. The view-specific classes are:

- .DESIGN
- .PREVIEW
- .OUTPUT

.OUTPUT is used when viewing the current section on the Live tab or in an external browser, and when generating output.

View selectors allow you to apply formatting to elements in a certain view, for example to highlight or show elements. The Designer itself does this, for example, to highlight all boxes in the Design view, when the Show Edges icon is clicked.

Adding an outline

The following style rule wraps every element that has the class address-block with a purple dashed outline in Design mode. The outline is not visible in other views or when outputting the document.

```
.DESIGN .address-block {
        outline: 1px dashed purple;
}
```

Adding a background pattern

The Postcard template wizard (in the Basic Print templates group) uses the .DESIGN class to mark areas that are reserved for postal use and should not contain text or images. These areas were added to the Master Page as absolute positioned boxes that have been given the class clearzone. The following style rule assigns a background pattern to elements with that class in the Design view:

```
.DESIGN .clearzone {
        background:url
(
AAAAFUlEQVQImWNgQAL/70z7TyqHgYEBANRfDcEzmlBaAAAAAElFTkSuQmCC)
repeat;
}
```

> **Note**
>
> The pattern image was created on www.patternify.com and is added as a data URI (see Data URIs).

Showing hidden Foundation elements

In Capture OnTheGo templates based on the Foundation framework the .DESIGN selector can be used to show elements that would otherwise be hidden in the Design view.
For example, to expand accordion elements and show validation errors in Design view, you could add the following style rules to your template:

```
.DESIGN .accordion .accordion-navigation > .content {
        display: block;
}
.DESIGN small.error {
        display: block;
        margin-top: -20px;
}
```

Loading a snippet via a script

It is possible, and often very useful, to load a snippet dynamically.

Create a script (see "Writing your own scripts" on page 393). In the code use the following function:

- For an HTML snippet: `loadhtml('snippets/nameofthesnippet.html')`.
  To insert the snippet in the content at any position where the script's selector is encountered, write `results.loadhtml('snippets/nameofthesnippet.html')`.
  Make sure that the file name is exactly the same as the file in the **Snippets** folder. If the file name isn't correct, the snippet will not appear in the template.
- For a JSON snippet: `loadjson('snippets/nameofthesnippet.json')`.
- For a Handlebars snippet: see "Handlebars in OL Connect" on page 452.

Tip

To insert the code to load a snippet even quicker, you can:

- **Drag a snippet into the Script window**. The function that loads the script - `loadhtml()` or `loadjson()`, depending on the file type - will automatically be added, including the file name.
- **Right-click a snippet** and select **Copy Resource Location** to copy the relative path of the snippet to the clipboard. It may then be pasted into a script.

Remote snippets are retrieved in the same way, except that the file extension should be `.rhtml` instead of `.html`. If it is a remote JSON snippet, the file extension is `.rjson`.
Again, note that the name of the snippet must be exactly the same as in the Snippets folder.

For more examples, see "loadhtml()" on page 824 and "loadjson()" on page 827.

Loading part of an HTML snippet

When a snippet contains a part that can be identified by a selector, that selector can be used to load that part of the snippet into a template.

In script, use the following code:

```
results.loadhtml('snippets/nameofthesnippet.html', 'selector')
```

See "loadhtml()" on page 824 for more information about this function.

Loading a snippet, depending on the value of a data field

To load a snippet depending on the value of a data field, you have to add a condition to the script.

Example

The following script evaluates if the value of the LANGUAGE field in the record is 'En'. If so, the snippet is added to the content.

```
if (record.fields.LANGUAGE == 'En') {
results.loadhtml('snippets/nameofthesnippet.html');
}
```

Another example is given in a how-to; see Load a snippet based on a data field value.

Loading part of a snippet, based on the value of a data field

When a snippet contains a part that can be identified by a selector, that selector can be used to load that part of the snippet into a template. It is possible to do this, based on the value of the data field. This is easiest when the selector matches the value of a data field.

Example

The following script reads the value of the LANGUAGE field in the record and uses that value as the selector in the function loadhtml(). If the snippet contains an HTML element with this ID (for example,), that HTML element will be added to the content:

```
var language = record.fields.LANGUAGE;
results.loadhtml('snippets/nameofthesnippet.html', '#'+ language)
```

Another example is given in the following how-to: Using a selector to load part of a snippet.

See also: "Standard Script API" on page 810.

Tip

An easy way to group content in an HTML snippet is putting each part in a container and giving that container an ID, for example:
This is text for English customers.

Use the function `.children()` to load the contents of the container, and not the container itself. For example:

```
results.loadhtml('Snippets/myfooter.html', '#EN').children()
```

This script loads the paragraph of the example (<p>), but not the container itself ( ).

Load a snippet and insert variable data into it

The following script loads part of an HTML snippet based on the value of a field, and then finds/replaces text with the value of a field before inserting the content into the document.

```
var promoTxt = loadhtml('snippets/promo-en.html', '#' +
record.fields['YOGA']);
promoTxt.find('@first@').text(record.fields['FIRSTNAME']);
results.html(promoTxt);
```

Loading content using a server's API

Content in a template is usually static (apart from being personalized) and part of the main text flow. It can also be located in a snippet (see "Snippets" on page 285).
It is also possible to include content that is served by another server. Many servers provide an API to fetch publicly available content from their site. That content may even be dynamic: the **most recent** blog posts on a Wordpress website, for example, or the **current** weather forecast for a certain city.
This topic explains how to retrieve content using a server's API and insert that content in a template.

Step 1: Getting the appropriate link

To request content from another server, you will need a link.
Some websites give the option to **embed** their content in your website by providing a link or the complete HTML. Youtube.com, for example, offers not only a link to share a certain video, but also the full HTML to embed that video in your website.
If that option is not available, you will have to build the link yourself. Find the server's **API** and look through it to get the exact endpoint and parameters that you need.

With many servers it is required to use an **API key** in the link; this key generally comes for free after you sign up to their website. The key will be part of the link that is used to make a request to the server.

Note

Pay attention to the service's Terms of Service. Many servers have limitations on the number of calls that can be made to them for free. Beyond these limits, their content will not show up in your template unless you purchase a business plan.

Step 2: Preparing the template

The next step is to set up a template. If you've got the HTML to embed content in your template you can paste that HTML on the Source tab (and skip Step 3).
Otherwise your template has to contain an element that can be replaced or followed by the remote content: an empty paragraph, for example, or a heading. If the element isn't unique in the template, give the element an ID.

Note that interactive content, such as an interactive map, can only be used in Web templates, and cannot be output on Print or Email contexts (even though they will show up in Preview mode!).

Step 3: Writing a script

The final step is to write a script that retrieves the content and inserts it into the template (see "Writing your own scripts" on page 393). Use the element or the ID of the element that you added in Step 2 as the script's selector. For information about selectors, see "Selectors in Connect" on page 414.

Tip

Select an element, then click on 'ID' in the Attributes pane, to create a script that has that element's ID as selector.

Retrieving content

Depending on the type of content that the remote server returns - HTML or JSON - you can use `loadhtml(location)` or `loadjson(location)` (see also: "loadhtml()" on page 824 and "loadjson()" on page 827) to retrieve the content. The link that you selected in Step 1 should be passed to the function as a string. For example:

```
loadjson('https://blog.mozilla.org/wp-json/wp/v2/posts?per_
page=5');
```

If the returned content is JSON data, that data has to be wrapped in HTML before inserting it into the template. This is demonstrated in the example below.

Tip

Install the Postman application to preview JSON returned by an endpoint.

Tip

To load a JavaScript file (.js) or a style sheet (.css) you can use `loadtext()`. See "loadtext()" on page 829.

Inserting content in the template

To insert the content after the selected element, use `results.after()`. To replace the element with the new content, use `results.html()` or `results.replaceWith()`.

Example: recent posts

The following script loads five posts from Mozilla's blog and inserts their titles as links in a template. Mozilla's blog is a WordPress website. Since the WordPress REST API uses JSON as the response format, the `loadjson()` function has to be used and the received content has to be wrapped in HTML.
If the script's selector was `h1` (a level one heading), the retrieved content would be inserted after each level one heading.

```
var postsObj = loadjson('https://blog.mozilla.org/wp-
json/wp/v2/posts?per_page=5');
var html = '';
html = '<ul>';
for (var idx in postsObj) {
        html += '<li><a href="' + postsObj[idx].link + '">' + postsObj
[idx].title.rendered + '</a></li>';
}
html += '</ul>';
results.after(html);
```

See WordPress REST API developer endpoint reference.

> **Tip**
>
> More examples of how to use an API to load external content are given in these How-to's:
>
> - Using the Google Maps API
> - Using the OpenWeatherMap API

Using scripts in Dynamic Tables

In most cases you don't need a script to change the content or style of rows or cells in a Dynamic Table.
However, if the contents or style of cells or rows in a Dynamic Table need to be personalized based on the value of a field, you will need a script.

This topic gives some information that will help you to write a script for (an element in) a Dynamic Table.
It is assumed that you are familiar with the scripting basics; see "Writing your own scripts" on page 393.

Quick-start a script with the Create script button

The easiest way to add a script that targets a specific element (usually a **cell** or **span**) that is linked to a data field in a detail table, is this:

1. In **Design** mode, select the element.
2. Click the **Create Script** button next to the **Field** drop-down on the **Attributes** pane.
3. Click **Yes**.

As a result:

- The element's `data-field` attribute will be renamed to `data-script`.
- If the element had a `data-format` attribute, that attribute will be removed.
- A new script will be created and opened in the Script Editor.

The **selector** of the new script is the `data-script` attribute with the name of the data field that the element was linked to as its value. For example: `table [data-script='ID']`.

The **code** of the script replaces the contents of the element with the current value of the data field:

```
var field = this.record.fields['ID'];
if (field) {
this.text(formatter.upperCase(field));
}
```

The **scope** of the script is set to **Each matched element** (see "Setting the scope of a script" on page 399). This means that in the code, `this` refers to the element that matches the selector, and `this.record` refers to the **current (nested) detail record** (see "this" on page 891).

> **Tip**
>
> To get access to the row in which the cell is located, you can use `this.parent()`.

> **Note**
>
> If you bind the element to a data field again (by selecting it and then selecting a field from the Field drop-down on the Attributes pane), the `data-script` attribute will be removed.

Writing scripts for a Dynamic Table

If you're going to create your own scripts for Dynamic Tables, there are a few things you need to know.

Selectors

Scripts that target (an element in) a Dynamic Table can use the same types of selectors as other scripts. In the output, Dynamic Table rows are repeated including any **classes** that are set on the row and on its contents.

In addition, Dynamic Tables and their rows and cells have some special **data- attributes** that can be used as selector. See: A Dynamic Table's data- attributes.
Note however, that the `data-repeat` attribute cannot be used as a selector for template scripts,

since it gets removed when the table is expanded, which happens before template scripts run. The `data-script` attribute is used if you let the Designer create a script for you (see "Quick-start a script with the Create script button" on page 426).

How the scope of a script can simplify code

It is recommended to set the scope of a script that targets (an element in) dynamically added rows to **Each matched element**.

The selector of a script that targets (something in) a row that is linked to a detail table will probably match multiple elements.
By setting the **scope** of a script you can determine whether you want to run the script **once**, or **once for each element** that matches the selector. (See "Setting the scope of a script" on page 399.)

If a script runs **once**, accessing the detail data and dynamically added rows tends to be bit complicated, especially when they are nested, since it involves looping over the `results` and `record` objects (see "results" on page 972 and "record" on page 848).

If a script targets (something in) a row that is bound to a detail table, and runs **once for each matched element**, the current element is accessible via the `this` object (see "this" on page 891) and the current detail record is accessible via `this.record`.

The advantage of the latter approach is demonstrated in the example below.

Example: Styling a row based on a value in a detail table

Here are two scripts that have the same effect: if in a row, the value of the field `Shipped` is 1, it colors the text of that row green.

Number	Description	Unit Price	Quantity
53674	Thule Crossroad Railing Foot Pack M450	199.95	3
62516	CamelBak Arete 18 Hydration Pack 2016 Black	65.00	1
117653	Swix Carving Kit 1	44.99	1
148080	Swix Alpine Racing Straps Ski Strap	9.99	4

The first script targets a **cell** in a Dynamic Table row that is bound to a detail table. The script has its scope set to "Each matched element", so it can use `this` (see "this" on page 891) to

access the selected cell and `this.record` to access the current detail record, without performing any loops.

Selector: `table [data-script='Shipped']`.
Scope: Each matched element

```
var field = this.record.fields['Shipped'];
if (field) {
        this.text(formatter.upperCase(field));
        if (field == 1)
                this.parent().css('color', 'green');
}
```

The following script targets the **table**, and has its scope set to "Result set". This script loops over the detail table in the `record`, evaluating the field `Shipped`. If the value of that field is 1, it looks up the corresponding row in the `results` (the selected Dynamic Table).

Selector: `#table_1 tbody`
Scope: Result set

```
for(var i = 0; i < record.tables.detail.length; i++){
       if(record.tables.detail[i].fields['Shipped'] == 1)
       query("tr:nth-child(" + (i+1) + ")", results).css
('color','green');
}
```

Control Scripts

When output is generated from a template, Control Scripts run **before** all other scripts, when a record is merged with a context. They determine how different sections of the context are handled. They can, for example, make the page numbering continue over all Print sections, split Email attachments, or omit Print sections from the output.

Some knowledge of JavaScript is needed to edit Control Scripts, just as for any other self-made scripts; see "Writing your own scripts" on page 393. Some functionality is provided as a Scripting Wizard, for example Conditional Print Sections.

This topic explains how to add a Control Script and it gives an overview of what Control Scripts can do. It will also tell you where you will find information about each feature, including examples.

The basics of script-writing in the Designer are explained in the following topic: "Writing your own scripts" on page 393.

What Control Scripts are

Control Scripts are a special kind of Designer script. They can manipulate the way output is generated from a template. They allow you, for example, to change the page numbering in Print output, to split one generated Print document into multiple Email attachments, or to set a Print section's background dynamically. (These are only a few examples; for more uses of Control Scripts see "What to use a Control Script for" on the next page.)

Control Scripts differ from Standard scripts in two ways:

- Control Scripts run **before** all other scripts. When a template consists of several contexts, and these contexts are combined in the output - for example, when an Email is generated with the Print context as attachment - all scripts run once for each context, but Control Scripts always go first.
- Control Scripts **do not touch the content** - meaning, the text flow - of the sections. They don't have a selector, like the other scripts do. A selector selects parts of the content of a section and stores them in the `results` object, so that they can be modified in the script. As Control Scripts don't have a selector, the `results` object can't be used there. Similarly, the `query()` function, which is used to select content from within a script, is unavailable in a Control Script.

Adding a Control Script

To add a Control Script:

1. On the **Scripts** pane at the bottom left, click the black triangle on the **New** button and click **Control Script**. A new script appears in the list.
2. Double-click the new script to open it. The script editor appears.
3. Change the name of the script so that it reflects what the script does.

> **Note**
>
> Scripts can only have the same name when they are not in the same folder.

4. Write the script; see the "Control Script API" on page 931. If you are not familiar with scripting, also see "Writing your own scripts" on page 393.

> **Tip**
>
> New Control Scripts added to the template contain code to continue the page numbering over all print sections, and two examples: one to select different sections of a Print context for email and print output, and one to select a Web section.

What to use a Control Script for

Control Scripts let you change the way a template is merged, by giving access to the template with all its contexts and sections in a script. A Control Script may, for example, omit, group and clone sections; add a background to a Print section; or add a header to an email. A number of the things that you can do with them is listed in the table below, with a link to a topic that explains how to do it and that shows what the script should look like.

In a Control Script, `section` usually is the most important object. To get a quick overview and lots of examples, see "section" on page 978. For help on specific tasks, see the table below.

Task	See topic	Field/function of `section` object
Change the **page numbering** of Print sections	"Control Script: Page numbering" on page 433	`restartPageNumbering`
Set the **background image** of a Print section	"Control Script: Setting a Print section's background" on page 438	`background.source`, `background.url`, `background.position`
Split and rename Print email **attachments**	"Parts: splitting and renaming email attachments" on page 436	`part`
Dynamically set a **password** on PDF attachments	"Control Script: Securing PDF attachments" on page 443	`password`, `ownerPassword`

Task	See topic	Field/function of `section` object
Include/exclude sections: • Conditionally omit Print or Email sections • Output one section or another, based on the value of a data field • Select one Print section as PDF attachment if the output is to be emailed, and another Print section if the output is to be printed.	Use a Conditional Print Section script to in-/exclude a Print section based on a simple condition; see "section" on page 978 In all other cases take a look at the examples in the following topic: "section" on page 978.	`enabled`
Copy and add sections dynamically	"Dynamically adding sections (cloning)" on page 440.	`clone()`
Add a **header** to an email	"section" on page 978, example: "Adding custom ESP handling instructions" on page 1023.	`headers`
Set the **margins** of a Print section or Master Page	"margins" on page 987	`margins`
Set a **Master Page**,	"sheetConfig" on page 954	`sheetConfig`

Task	See topic	Field/function of `section` object
Media, or **Duplex** printing for a Print section		
Repeat the sheet configuration after a number of pages	"sheetConfig" on page 954	`sheetConfig.positions.repeat`
Dynamically change the **page size** of print sections and media	"section" on page 978, "media" on page 937 **Note:** When a Control Script changes the size of a section, it should also change the size of the linked Media; this is not done automatically. While the output may still look good, a size mismatch can cause an issue if a script or another component assumes that the media size matches the section size. In case of a size mismatch a preflight will show a warning (see "Doing a Preflight" on page 405).	`height`, `width`

Control Script: Page numbering

This topic explains how to write a Control Script that changes the page numbering in Print sections. Note that when you add a Control Script, it already contains a script to make the page numbering continue over all Print sections.

For information about Control Scripts in general, see "Control Scripts" on page 429 and "Control Script API" on page 931. If you don't know how to write scripts, see "Writing your own scripts" on page 393.

How to change page numbering in a control script

A Control Script can make the page numbering continue over all Print sections or let it restart on a section. This is done by setting the `restartPageNumber` field on a `section` to `true` or `false`. For example: `merge.template.contexts.PRINT.sections['Section 2'].restartPageNumber = true;`. (Also see "section" on page 978 and "Control Script API" on page 931.)

Page numbering starts with page 1 for each section. If for a section `restartPageNumber` is set to `false`, that section will start with the page number following the last page of the previous section.

Note that even if a section is not enabled (so it will not be outputted), its `restartPageNumber` flag is still taken into account for composing the page number sequences.

By default, each section has `restartPageNumber = false` when the first control script runs.

> **Tip**
>
> If you are looking to create a short, simple table of contents in **one section**, you could add a Standard Script that uses the `pageRef()` function. For an example, see "pageref ()" on page 919.
> For a **multi-page**, **cross-section** table of contents you must use a Post Pagination Script; see "Creating a Table Of Contents" on page 447.

Examples

Restarting the page numbers several times

Assume that a template has four sections (of 1 page each) in the Print context and a Control Script sets the page numbering as follows:

1. Section A (1 page) restartPageNumber = true
2. Section B (1 page) restartPageNumber = true

3. Section C (1 page) restartPageNumber = false
4. Section D (1 page) restartPageNumber = true

The code would look like this:

```
if (merge.context.type == ContextType.PRINT) {
        merge.context.sections['Section A'].restartPageNumber = true;
        merge.context.sections['Section B'].restartPageNumber = true;
        merge.context.sections['Section C'].restartPageNumber = false;
        merge.context.sections['Section D'].restartPageNumber = true;
}
```

The page numbering in the output will be:

1. Section A page 1
2. Section B page 1
3. Section C page 2
4. Section D page 1

Disabled section

When a section is disabled, it will not be outputted, but its `restartPageNumber` flag will still be taken into account for composing the page number sequences. So, if the `restartPageNumber` flags are set as follows:

1. Section A (1 page) restartPageNumber = true
2. Section B (2 pages) restartPageNumber = false
3. Section C (3 pages) restartPageNumber = true, enabled = false
4. Section D (4 pages) restartPageNumber = false

In code:

```
if (merge.context.type == ContextType.PRINT) {
        merge.context.sections['Section A'].restartPageNumber = true;
        merge.context.sections['Section B'].restartPageNumber = false;
        merge.context.sections['Section C'].restartPageNumber = true;
        merge.context.sections['Section C'].enabled = false;
        merge.context.sections['Section D'].restartPageNumber = false;
}
```

The page numbering in the output will be:

1. Section A page 1
2. Section B page 2
3. Section D page 1 (page numbering is restarted due to section C's restartPageNumber = true)

Parts: splitting and renaming email attachments

In a Control Script, **parts** can be defined to determine which sections should be output to the same file. This way it is possible to split the Print context into multiple email attachments. This topic shows how to do that.

For information about Control Scripts in general, see "Control Scripts" on page 429 and "Control Script API" on page 931. If you don't know how to write scripts, see "Writing your own scripts" on page 393.

Defining parts

Defining parts is done by setting the `part` field on a `section`, for example: `merge.template.contexts.PRINT.sections['Section 2'].part = "PDF_Attachment2";`. (Also see "section" on page 978 and "Control Script API" on page 931.)

- If a part name is given, then that delimits the start of a new part (even if the part name is the same as the previous one). Following sections that don't define a part name, will be added to the previous part.
- A part ends at the last enabled* section or at the last section before the start of a new part. *When a Control Script has set the `enabled` field of a `section` to `false`, it will not be outputted.

If no part name is set on any section, it is assumed that there is only one part, consisting of all Print sections. The attachment will be named after the email subject.

Examples

No parts defined

Assume there are three Print sections: sections A, B and C. When generating Email output with the Print context as attachment, all three Print sections will be put together in one file and

attached to the email. If the email's subject is 'Take action', the name of the attached file will be 'Take action.PDF'.

Splitting and renaming a Print attachment

Assume there are three Print sections: sections A, B and C. In a Control Script a part name is defined for section C:

```
var section = merge.template.contexts.PRINT.sections['Section C'];
section.part = 'Part2';
```

When generating Email output with the Print context as attachment, the email will have two attachments:

- attachment 1: Section A, Section B
- attachment 2: "Part2", which is Section C. The file name of this attachment is the part name.

Controlling multiple Email attachments

The following script attaches the following sections to an email:

- Print section 3 + 4 as attachment with continued page numbers
- Print section 6 as separate attachment

```
if (channel == Channel.EMAIL) { // only when generating Email
output
if (merge.context.type == ContextType.PRINT) {
        merge.context.sections['Section 1'].enabled = false;
        merge.context.sections['Section 2'].enabled = false;
        merge.context.sections['Section 3'].enabled = true;
        merge.context.sections['Section 3'].part = "PDFAttach1";
        merge.context.sections['Section 4'].enabled = true;
        merge.context.sections['Section 4'].restartPageNumber = false;
        merge.context.sections['Section 5'].enabled = false;
        merge.context.sections['Section 6'].enabled = true;
        merge.context.sections['Section 6'].part = "PDFAttach2";
}
}
```

Note

For another example, see this how-to: Output sections conditionally.

Control Script: Setting a Print section's background

In the Print context, an image file can be used as a Print section's background; see "Using a PDF file or other image as background" on page 142.

If you want the section background to be switched automatically, depending on the value of a data field, you need a Control Script. There is a Script Wizard that can generate that script for you, provided that certain conditions are met; see: "Dynamic Print section backgrounds" on page 368.
Otherwise, you will have to write the Control Script yourself. This topic explains how to write a Control Script that sets a Print section's background.
Note that the settings made in a Control Script take precedence over the settings made in the Print Section Properties dialog.

Note

Encrypted PDF files are **not supported** in *PDF pass-through* mode.

For information about Control Scripts in general, see "Control Scripts" on page 429 and "Control Script API" on page 931. If you don't know how to write scripts, see "Writing your own scripts" on page 393.

Setting a background in script

The Control Script should first enable a background on the section, in case an initial background wasn't set via the user interface. This is done by setting the source type for the background of the section to Resource PDF (see "BackgroundResource" on page 962). For example:

```
merge.template.contexts.PRINT.sections['Policy'].background.source
= BackgroundResource.RESOURCE_PDF;
```

For a Resource PDF, the Control Script should specify a path, for example:

```
var resourceUrl = 'images/policy-' + record.fields.policy + '.pdf';
merge.template.contexts.PRINT.sections['Policy'].background.url =
resourceUrl;
```

Note

An image with an unknown file extension is represented by a red cross in the output, but no error is logged unless the image refers to a local file that does not exist on disk.
Image file extensions that the software recognizes are: AFP, BMP, EPS, GIF, JPG/JPEG, PCL, PDF, PNG, PS, SVG, and TIF/TIFF.

Positioning, scaling and rotating the background

After a background has been selected, it can be positioned, scaled and rotated, using properties of the background object; see "background" on page 951.

To position the background, for example, set the section's background.position:

```
activeSection.background.position = MediaPosition.FIT_TO_MEDIA;
```

For all possible positions, see "MediaPosition" on page 966.

Setting a page range in script

When a PDF that serves as a dynamic section background has multiple pages, you can specify a range of pages to be used, in a control script.
Put the number of the first page in the range in the section's background.start field and the last page in background.end.

The following script sets the page range from 2 to 5:

```
merge.template.contexts.PRINT.sections['Policy'].background.start =
2;
merge.template.contexts.PRINT.sections['Policy'].background.end =
5;
```

Setting a page range automatically sets background.allPages to false (see "background" on page 951).
On the other hand, when you first define a page range and then set background.allPages to true, this disables the page range.

Tip

You could use the `resource()` function to check the number of pages or for example the page height and width before setting it as a background (see "resource()" on page 940).

Example

This scripts sets a background on a Print section using absolute positioning.

```
var activeSection = merge.template.contexts.PRINT.sections['Section
1'];
activeSection.background.source = BackgroundResource.RESOURCE_PDF;
activeSection.background.url = "images/somepage.pdf";
activeSection.background.position = MediaPosition.ABSOLUTE;
activeSection.background.left = "10mm";
activeSection.background.top = "10mm";
```

You could replace the last three lines of the previous script by the following line to scale the Print section background to Media size:

```
activeSection.background.position = MediaPosition.FIT_TO_MEDIA;
```

Dynamically adding sections (cloning)

This topic explains how to "clone" (copy) a section in a Control Script. Print sections can be cloned, so that a document can have a dynamic number of sections, based on data. This is particularly useful when the record set defines one or more PDFs (e.g. insurance policies) per recipient. Via a Control Script, for each PDF a section can be cloned and each clone can be given one of the PDFs as background (see "Control Script: Setting a Print section's background" on page 438). For each page in the PDF, a page will be added to the section.

For information about Control Scripts in general, see "Control Scripts" on page 429 and "Control Script API" on page 931. If you don't know how to write scripts, see "Writing your own scripts" on page 393.

Note

Clones are only visible in the output, not on the Preview tab.

Cloning a section

To clone a section, first use the `clone()` function and then add the clone to the Print context before or after a specific section, using `addAfter()` or `addBefore()`:

```
var printSections = merge.template.contexts.PRINT.sections;
var clone = printSections["Section 1"].clone();
printSections["Section 1"].addAfter(clone);
```

Cloned sections have the same properties as normal sections (see "section" on page 978), but they cannot call the section functions `clone()`, `addBefore()` and `addAfter()`, which means you cannot clone a clone, or use a clone as a starting point to insert other clones into a template.

Note that with multiple clones, the next clone is always added **after** the previous clone.
With `addBefore()`, the code `original.addBefore(clone1); original.addBefore(clone2);` will result in "clone1, clone2, original".
With `addAfter()` the code `original.addAfter(clone1); original.addAfter(clone2);` results in "original, clone1, clone2".

Renaming a clone

By default, clones receive the name of their source section with a "Clone {unique identifier}" suffix, for example:
Source: "Section 1"
Clone Name: "Section 1 Clone 71c3e414-fdf3-11e8-8eb2-f2801f1b9fd1"

Use the `name` property to assign another name to the cloned section, for example:

```
clone.name = "my_section_clone";
```

Just like section names, the clone's name should be unique - within the scope of a single record, that is; across records, the same name can be used.
When two clones get the same name in the same record, one of the clones may no longer be used. This is prevented by giving clones unique names. Unique names also ensure that CSS rules and scripts can target a specific clone.

Targeting elements in a cloned section

Clones that have a unique name can be further personalized with the use of CSS style sheets (see "Styling and formatting" on page 294) and personalization scripts (see "Variable Data" on page 351 and "Writing your own scripts" on page 393).
The selector to use is: [section="*name of the clone*"].

The following CSS style rules target the <h1> element in a number of clones and assign the respective text a different color:

```
[section="my_section_clone_0"] h1 { color: red; }
[section="my_section_clone_1"] h1 { color: green; }
[section="my_section_clone_2"] h1 { color: blue; }
```

The same selector could be used in personalization scripts:

Selector: [section="my_section_clone_0"] h1
Script: `results.css('color','red');`

Inside a Standard Script, cloned sections can be found using `merge.section`:

```
if (merge.section == "my_section_clone_0") {
        results.html("Clone!");
} else {
        results.html("Original.");
}
```

Note that in a Control Script, `merge.section` is only defined when the output channel is WEB; see "merge" on page 989.

Examples

Cloning a section based on the number of records in a detail table

This script creates as many clones of a section as there are records in a detail table. It assigns the new sections a unique name.

```
var printSections = merge.template.contexts.PRINT.sections;
var numClones = record.tables['detail'].length;
for( var i = 0; i < numClones; i++){
       var clone = printSections["Section 1"].clone();
       clone.name = "my_section_clone_" + i;
       printSections["Section 1"].addAfter(clone);
}
```

Cloning a section based on data and assigning a background PDF

This script clones a section based on data fields. It disables the source section first and then calls the addPolicy function. addPolicy clones the section, renames it and sets a PDF from the

resources as its background. It explicitly enables the clone and then adds it to the Print context.

```
var printSections = merge.template.contexts.PRINT.sections;
merge.template.contexts.PRINT.sections["Policy"].enabled = false;
if(record.fields.policy_a == 1) {
	addPolicy('a');
}
if(record.fields.policy_b == 1) {
	addPolicy('b');
}
function addPolicy(policy){
	var resourceUrl = 'images/policy-' + policy + '.pdf';
	var clone = printSections["Policy"].clone();
	clone.name = "policy_" + policy;
	clone.background.url = resourceUrl;
	clone.enabled = true;
	printSections["Policy"].addAfter(clone);
}
```

Control Script: Securing PDF attachments

The Print context can be attached to an email in the form of a PDF file and secured with a password. This can be done without a Control Script, see "Email attachments" on page 187 and "Email PDF password" on page 186.
With a Control Script, you can do the same, and more: the attachment can be split into multiple attachments (see Parts). Each attachment may have a different (or no) set of passwords, so you could mix secured and unsecured attachments in a single email. This topic shows how.

For information about Control Scripts in general, see "Control Scripts" on page 429 and "Control Script API" on page 931. If you don't know how to write scripts, see "Writing your own scripts" on page 393.

Setting passwords in script

To set a password on a Print section in a Control Script, the script should first retrieve the Print section/s using `merge.template.contexts.PRINT.sections` or `merge.context.sections` (also see the example below).
Next, the script can split the attachments, if needed (see "Parts: splitting and renaming email attachments" on page 436), and it can set a password on each `section`. For example:

- `merge.template.contexts.PRINT.sections['Section 2'].password = 'secret';`
- `merge.template.contexts.PRINT.sections['Section 2'].ownerPassword = 'secret';`

When producing a **single** attachment, the password(s) should be set on the first Print section. When producing **multiple** attachments, it should be set on the first section of each part.

Password types

PDF allows for two types of passwords to be set on a secured PDF file: a user password and owner password. The user password allows a limited access to the file (e.g. printing or copying text from the PDF is not allowed). The owner password allows normal access to the file. The Email PDF password script sets both the user and owner password to the same value, so that when the recipient provides the password, he can manipulate the file without limitations.

In a Control Script:

- `password` is used to set the user password and owner password for a PDF attachment to the same value.
- `ownerPassword` is used to set the owner password for a PDF attachment. Setting only the owner password creates a secured PDF that can be freely viewed, but cannot be manipulated unless the owner password is provided. Note that the recipient needs Adobe Acrobat to do this, because the Acrobat Reader does not allow users to enter the owner password.

Removing a password

Passwords set in the Control Script override the password set through the Email PDF password script (see "Email PDF password" on page 186). This allows you to change or remove the password from a specific part. Removal is done by setting the `password` field to `null` or an empty string ("").

Example

This scripts splits the Print output into two PDF attachments and sets a password for the second attachment.

```
var printSections;
if (channel == Channel.EMAIL) { // only when generating Email
output
if (merge.context.type == ContextType.PRINT) {
```

```
printSections = merge.template.contexts.PRINT.sections;
        printSections['Section 1'].part = 'PDFAttach1';
        printSections['Section 2'].part = 'PDFAttach2'
        printSections['Section 2'].password = 'secret';
        }
}
```

Post Pagination Scripts

Post Pagination Scripts are run in a **Print** context **after** the content has been paginated. Because they can search through the output of all Print sections, and modify Print sections (one at a time), they may be used to create a Table Of Contents (TOC), as explained in the topic: "Creating a Table Of Contents" on page 447.

This topic explains what a Post Pagination Script is and how to add it to a template.

The basics of script-writing in the Designer are explained in the following topic: "Writing your own scripts" on page 393.

What Post Pagination Scripts are

Post Pagination Scripts are a special kind of Designer script: they are applied to the output of all sections in a **Print** context **after** the content has been paginated.

Post Pagination Scripts differ from Standard Scripts and Control Scripts in two ways:

- Post Pagination Scripts run **after** all other scripts, more precisely: after the content has been *paginated*. The pagination process applies page breaks to the content of a Print section, adds Master Pages and sets the Media. A Post Pagination Script may query the rendered document and collect information about elements (for instance, on which page they reside on) and sections (for instance, whether they are enabled). If needed, a Post Pagination Script can change the sheet configuration and re-paginate a section.
- Post Pagination Scripts only apply to the **Print** context. The output of other contexts is not paginated.

Just like Standard Scripts, Post Pagination Scripts have a **selector** (see: "Selectors in Connect" on page 414). A selector selects parts of the content of a section and stores them in the `results` object, so that they can be modified in the script (see "results" on page 972).

The second most important object in a Post Pagination Script, just like in a Control Script, is a `section` (see "section" on page 978).

What to use a Post Pagination Script for

After all Print sections have been paginated, Post Pagination Scripts may search through the rendered document and collect information about elements (for instance, which page they reside on) and sections (for instance, whether they are enabled). With this information, a Post Pagination Script can do two things:

- **Modify the output**. The script may modify the output of a section. It could, for example, use information like page numbers to create a Table Of Contents (TOC), as explained in the topic: "Creating a Table Of Contents" on the next page. If needed, a Post Pagination Script can change the sheet configuration and re-paginate the section (see "paginate()" on page 988.
- **Add information to the Connect database**. The script may add production information, such as the page, size or position of elements after a merge, as custom properties to a Print Content Item in the Connect database (see "contentitem" on page 968). Custom properties can be utilized for further processing in a Workflow configuration with the Retrieve Items task. The Retrieve Items task retrieves custom properties along with the base record information (see Retrieve Items in Workflow's Online Help).

Adding a Post Pagination Script

To add a Post Pagination Script:

1. On the **Scripts** pane at the bottom left, click the black triangle on the **New** button and click **Post Pagination Script**. The new script appears in the Post Pagination folder.
2. Double-click the new script to open it. The script editor appears.
3. Change the name of the script so that it reflects what the script does.

> **Note**
>
> Scripts can only have the same name when they are not in the same folder.

4. Write the script. If you are not familiar with scripting, see "Writing your own scripts" on page 393.

Note

Post Pagination Scripts run only when a Print *section* is previewed or outputted.
To verify the results of Post Pagination Scripts on a certain *Master Page*, preview the Print section to which that Master Page is applied.

Creating a Table Of Contents

This topic explains how to create a multi-page, cross-section Table Of Contents (TOC) using a Post Pagination Script.
For information about Post Pagination Scripts in general, see "Post Pagination Scripts" on page 445.
The basics of script-writing in the Designer are explained in the following topic: "Writing your own scripts" on page 393.

Step 1: Opening a Print template

Create or open a Print template.
Make sure to use HTML headings level 1 and level 2 (**<h1>** and **<h2>**) in your Print sections, if you want to use the script sample that is given in Step 3. This script collects the text and page numbers of these headings and puts them in the TOC.
To quickly change a paragraph into a Heading, place the cursor inside of it, or select the paragraph (see: "Selecting an element" on page 196). Then select the appropriate element, either on the **Format** menu, or from the 'Element type' drop-down on the toolbar.
Of course, you could just as well create a table of contents using other heading levels, or even other elements. In that case you'll have to adjust the script accordingly.

Step 2: Creating a placeholder for the TOC

Create one extra Print section to put the Table Of Contents in.
Inside the extra section, insert an Article element (see "Inserting an element" on page 194).
Give the element an ID, for example: toc-content. This element is the placeholder for the TOC.

Step 3: Inserting the Post Pagination script

Insert a Post Pagination script (see "Adding a Post Pagination Script" on page 446).
Double-click on the new script to open it.
Set its **selector** to the ID that you specified in Step 2, preceded by a #, for example: #toc-content.
Paste the following code in the script editor:

```
// Build the TOC
var toc = '';
merge.context.query("h1, h2").each(function() {
        var pageNo = this.info().pageNo;
        var text = this.text();
        var level = this.tagName();

        toc += '<p class="toc-entry ' + level + '">';
        toc += '<span class="text">' + text + '</span>';
        toc += '<span class="dots"></span>';
        toc += '<span class="number">' + pageNo + '</span></p>';
});

results.html( toc );

// Repaginate the result
merge.section.paginate();

// Update the page numbers
var $numbers = query('.number');
merge.context.query("h1, h2").each(function( index ) {
        var pageNo = this.info().pageNo;
        var entry = $numbers.get( index );
        if( entry ) {
                entry.text( pageNo );
        }
});
```

What the script does

First the script creates a variable to hold the table of contents: toc.
Then it collects all <h1> and <h2> elements - in other words, level 1 and 2 headings. The merge.context.query(selector) function searches **across all Print sections** (see "query (selector)" on page 971).

The query returns a result set. Each of the elements in the result set goes through the callback function defined in each() (see "each()" on page 813).

The callback function gets the element's page number, text and HTML tag name:

```
        var pageNo = this.info().pageNo;
        var text = this.text();
        var level = this.tagName();
```

Note that the info() function can also be used to get an element's sheet number, the section it is located in, and the total page count and sheet count of that section (see "PaginationInfo" on page 971 and "info()" on page 977). In this case only the page number is used.

Then the callback function adds an entry to the variable that holds the table of contents, using the retrieved info.

```
toc += '<p class="toc-entry ' + level + '">';
toc += '<span class="text">' + text + '</span>';
toc += '<span class="dots"></span>';
toc += '<span class="number">' + pageNo + '</span></p>';
```

The HTML tag name is added as a **class**. This can be used in a CSS file to style the entries in the table of contents according to their level. (See "Step 4: Styling the table of contents" on page 451.)
The empty span between the heading's text and page number has the class dots. This is used to put dots between heading and page number.
The number class (for the page number) is not only used in CSS, but also later on in the script.

The table of contents is inserted in the section with: results.html(toc); (see "html()" on page 915).

The table of contents may get too long for a single page and affect the page numbers in other sections. In that case it is necessary to re-paginate the content; merge.section.paginate(); does the trick.

> **Note**
>
> Whether page numbering restarts in each section, depends on the settings for page numbering; see "Configuring page numbers" on page 152. By default, page numbering starts with page 1 for each section.

If the pagination process has changed the page numbers, the TOC needs to be updated as well.
To do that, the script first has to collect the page numbers from the table of contents. This is where the number class comes in handy: var $numbers = query('.number');.
Note that this query() function, as opposed to merge.context.query(), only searches the **current** section (see "query()" on page 845).
Then, the level 1 and 2 headings are collected from all Print sections again using

merge.context.query("h1, h2").
The callback function in each() retrieves the heading's new page number. It then uses the index number of the heading in the result set to get the corresponding entry in the TOC: var entry = $numbers.get(index); (see "get(index)" on page 873), and replaces it with the new page number.

Excluding headings

Often there are certain headings that you don't want to appear in the table of contents. The title of the table of contents itself, for example.
To exclude these headings from the table of contents, do the following:

1. Give all the headings that you want to be ignored the same class (see "ID and class" on page 193), for example `ignore-me`.
2. Add the `:not` selector to the queries in the script, specifying that class. Remember to put a dot before the name of the class. For example: `merge.context.query("h1:not(.ignore-me), h2").`

Adding internal hyperlinks

It is possible to create links in a table of contents that point to the respective position in the document (for example when generating PDF output). An internal hyperlink points to an element with a specified ID. (See: "Hyperlink and mailto link" on page 268.)

Here's how to change the each() function in the script in order to add internal hyperlinks to the table of contents:

1. Add the following variable: `var linknumber = 1;`. This variable is used to generate unique IDs for elements that don't have an ID.
2. Before the first line that starts with 'toc ', add the following:

```
var anchorId = '';
if (this.attr("id")) {
anchorId = this.attr("id");
}
else {
anchorId = 'generatedID-' + linknumber;
linknumber++;
this.attr("id", anchorId);
}
```

This code takes the element's ID. If an element doesn't have an ID, the script generates a new, unique ID for it.

3. Replace the line:
   `toc += '<span class="text">' + text + '</span>';`
   with this line:
   `toc += '<a class="text" href="#' + anchorId + '">' + text + '</a>';`
   The new line adds a hyperlink: <a ... href> with the element's ID to the table of contents.

Step 4: Styling the table of contents

Each component of the table of contents inserted in Step 3 is wrapped in a element that has a class, for example: "dots" or even multiple classes: "toc-entry h1". (Class names are separated by a space.)
These classes can be used to style the table of contents with CSS. The principles of styling with CSS are explained in another topic: "Styling templates with CSS files" on page 296.

Add the following **styles** to the style sheet for the Print context (see "Step 1: edit CSS" on page 300) to align the page number to the right and fill the space between the text and the page number with leader dots.

```
p.toc-entry {
display: flex;
margin: 0 0 0.25em 0;
}

p.toc-entry .text {
flex-shrink: 0;
}

p.toc-entry .dots {
flex-shrink: 1;
white-space: nowrap;
overflow: hidden;
text-overflow: clip;
}

p.toc-entry .dots::after {
font-weight: normal;
content: ". . . . . .
. . . . . .
```

```
. . . .
```

Note

The information in this Online Help focuses on the implementation of Handlebars in OL Connect. For general information about Handlebars and how to use it, see the following web sites: https://handlebarsjs.com/ and https://devdocs.io/handlebars.

Handlebars expressions

Note

The information in this Online Help focuses on the implementation of Handlebars in OL Connect. For general information about Handlebars and how to use it, see the following web sites: https://handlebarsjs.com/ and https://devdocs.io/handlebars.

Data in expressions

A handlebars expression is a **{{**, some contents, followed by a **}}**.
For example:
Hello {{firstname}} {{lastname}}!

In its most simple form the contents of the expression refer to a data field. They will be replaced with the value of that field if it exists in the data that is passed to the template (see "Render the Handlebars template" on page 292).

You can simply **write** the field or key name in the expression. A whitespace before and after the field or key name is accepted, e.g. {{ fullName }}.

Another, quick way to insert a simple expression is to **drag and drop** the field from the Data Model in the Handlebars template.

When you drag and drop a field into a Handlebars template, a call to a function that formats the value may be added automatically. This depends on the field type.
For example, dragging a currency field will produce: {{currency fieldName}} which outputs a number formatted as an amount of money.
You can remove the function or change it (see "Formatting values" on the next page).

When a data field of the type **HTML String** is dragged into a Handlebars snippet, the expression gets three curly brackets to ensure that HTML characters are not escaped (see "HTML values" on the next page).

It is possible to use logic in an expression, for example to display content only if a field has a certain value. How this is done is described in another topic: "Using logic in a Handlebars template: helpers" on page 456.

> **Note**
>
> In OL Connect (as opposed to the original Handlebars library), numbers in Handlebars expressions do not need to be surrounded by quotes. For example, to calculate 1+2 you can write (add 1 2) instead of (add "1" "2").

Accessing data at different levels

A field or key name without a **.** or **/** is assumed to refer to a field or table at the root of the data with which the template is rendered. There are a number of ways to access data that is not located at the root.

- With **.** or **/** you can navigate **down** into the record or object. For example: {{Cars.Brand}} and {{Cars/Brand}} refer to the same field.
  Note that to access an item in a detail table or array directly, you also have to navigate down. For example: Cars.Brand.Models.[0] refers to the first item in Models.
- **@root** represents the top level of the data.
  If a template is rendered with the current record or part of it, @root refers to the root of the current record.
  With a JavaScript object, @root refers to the root level of the data that was passed to the Handlebars template.
- The built-in helpers #each and #with dive into an object-property, giving you direct access to its properties. (See "Using logic in a Handlebars template: helpers" on the facing page and the documentation of Handlebars.)
- Use **../** to navigate one level **up**. This is particularly useful in #each and #with blocks.

> **Note**
>
> If a template is rendered with part of the current record, the template still has access

to the *entire* record and can navigate up and outside of that part, to a higher level in the current record.
If a JavaScript object is passed, the template only has access to the passed data, and it does not have access to the current record.

Formatting values

Functions that can be called from a Handlebars expression are called *helpers*. OL Connect provides a number of helpers to **format the value** that is the result of an expression. For example: {{upperCase firstname}} returns the value of the firstname field in uppercase.

The available functions are the same as those provided by the `formatter` object in standard Designer scripts; see "formatter" on page 817.
Note that the names of these functions are case sensitive. For example, "upperCase" works, but "uppercase" does not.

The values will be formatted according to the rules of the current locale. See "Locale" on page 336.

HTML values

The value returned by an {{expression}} is HTML-escaped. This means that those characters that have a special function in HTML: **& < > \" ' ` =** are replaced with HTML character references. For example: **&** will be changed into **&**.

Take this expression: {{foo}}
If the field foo contains a '>' and you don't HTML-escape it, this would produce invalid HTML: >
After HTML-escaping, it becomes valid HTML: >

If the result of an expression should not be HTML-escaped, the expression must be enclosed in three curly brackets: **{{{** ... **}}}**.

When an HTML field is dragged into a Handlebars template it is automatically enclosed in three curly brackets.

Note

Content provided by 'block helpers' - expressions in the form {{#helperName arguments}}...{{/helperName}} - is **not** automatically HTML-escaped. See "Using logic in a Handlebars template: helpers" below.

Using logic in a Handlebars template: helpers

Note

The information in this Online Help focuses on the implementation of Handlebars in OL Connect. For general information about Handlebars and how to use it, see the following web sites: https://handlebarsjs.com/ and https://devdocs.io/handlebars.

Block helpers

Functions that can be called from a Handlebars expression are called *helpers*. Handlebars offers a number of built-in *block helpers* that let you use logic in templates. Block helpers look like this: {{#helperName arguments}}...{{/helperName}}.

They are documented on Handlebars' website: https://handlebarsjs.com/guide/builtin-helpers.html.

Of those helpers, the following are supported in OL Connect:

- **#if**: Conditionally renders a block. An optional {{else}} section inside the block will display when the condition is not true.
- **#unless**: Renders a block if the expression returns a falsy value (i.e. a value that is considered false when encountered in a Boolean context). An optional {{else}} section inside the block will display when the condition is true.
- **#each**: Allows to iterate over a list. An optional {{else}} section inside the block will display when the list is empty.
- **#with**: Can be used to work directly with the passed object or object property. An optional {{else}} section will display only when the passed value is empty.

In block helpers, the data variables **@first**, **@last**, **@index** can be used; see https://handlebarsjs.com/api-reference/data-variables.html.

Unlike in the original Handlebars library, content provided by a block helper is **not automatically HTML-escaped** in OL Connect. Blocks are typically used to generate HTML, so it is assumed that the result consists of well-formed HTML.

> **Note**
>
> An empty array is considered falsy (i.e. evaluated as false). This is normally not the case in JavaScript, but the Handlebars library makes an exception.
>
> The `lookup` helper and `log` block helper are **not** implemented in OL Connect.
>
> The data variable **@key** is not supported in OL Connect. OL Connect only supports iterating over arrays and tables, not over arbitrary objects.
>
> Since content provided by a block helper is not HTML-escaped in OL Connect, Handlebars' `SafeString` class is not needed and therefore not supported.

Examples of block helpers

#if

The following #if block will output a table row with two cells if the data field named O_L10095 is not false, undefined, null, 0, an empty string, or an empty array.

```
{{#if O_L10095}}
<tr>
        <td>Collective insurance</td>
        <td>{{O_L10095}}</td>
</tr>
{{/if}}
```

#unless and #each

The following Handlebars template outputs a row with a header and a variable number of clause descriptions, based on a detail table called 'Clause', *unless* there are no clauses, in which case the row gets a header and one cell that displays the text 'None'.

```
<table class="policy-table">
<tbody>
{{#unless Clause}}
```

```
<tr>
        <th class="h3">Clause(s)</th>
        <td>None</td>
</tr>
{{else}}
        {{#each Clause}}
        <tr>
                <th class="h3">{{#if @first}}Clause(s){{/if}}</th>
                <td>{{Descr}}</td>
        </tr>
        {{/each}}
{{/unless}}
</tbody>
</table>
```

#with and #each

This #with block calls a custom helper (see "Adding custom helpers" on the next page) that returns an object with the properties: object, amount, coverage and premium. The object's properties are accessed directly inside the block. The #with block is executed for each of the records in a detail table called 'insurance_coverage'.

```
{{#each insurance_coverage}}
        {{#with (getObjectAndCoverage)}}
        <tr>
                <td>{{object}}</td>
                <td class="curr">€</td>
                <td>{{amount}}</td>
                <td>{{coverage}}</td>
                <td class="curr">€</td>
                <td>{{premium}}</td>
        </tr>
        {{/with}}
{{/each}}
```

Additional helpers in OL Connect

OL Connect provides the following additional helpers.

- `eq`, `neq` (equal, not equal) - tests two values for equality
- `gt` (greater than), `gte` (greater than or equal), `lt` (lower than), `lte` (lower than or equal) - compares two numbers

- `not`, `or`, `and` - logical operators
- `add` - adds two numbers
- `sub` - subtracts two numbers
- `startsWith` - tests if a string starts with a certain prefix
- `endsWith` - tests if a string ends with a certain suffix
- `matches` - tests if a string matches a regular expression

These can be used **as expression** (returns the result). For example:
Recurs every month: {{eq 'Monthly' recurrence}}

They can also be used **in sub expressions**, for example of a block helper:
 {{#if (eq 'Monthly' recurrence)}} Recurs every month {{/if}}

Adding custom helpers

In both Handlebars and OL Connect, a number of helpers are predefined, but you can also create your own helpers using `Handlebars.registerHelper()`. See the Handlebars guide: https://handlebarsjs.com/guide/#custom-helpers.

> **Tip**
>
> Define any helpers in a Control Script. Control Scripts are executed first. See "Control Scripts" on page 429.

> **Note**
>
> The `blockHelperMissing` and `helperMissing` hooks are not implemented in OL Connect.
>
> Passing `options.hash`, `options.helpers` and `options.partials` arguments to helpers is not supported in OL Connect. (`options.data`, `options.fn` and `options.inverse` are supported.)
>
> Registering multiple helpers with a single call is not supported in OL Connect.

Examples of custom helpers

Date

The following code registers a function that returns the current date.

```
Handlebars.registerHelper('now', function() { return new Date();
});
```

After the helper is registered, the expression {{dateLong now}} in a Handlebars template will return the current date, formatted as a long date, e.g. February 23, 2022.

Get coverage

This helper returns an object whose properties are filled with values based on data fields in the current context (`this`). For example, when called within an #each block that is executed with the records of a detail table, `this` refers to the current detail table record. (See also: "#with and #each" on page 458.)

```
Handlebars.registerHelper('getObjectAndCoverage', function () {
        let coverage = this.O_L99042;
        let category = this.C_L99042
        if( category === 'extra' ) {
                        coverage = 'Extra + glass';
        }
        return {
                "object": "Residential building",
                "amount": this.L10039D,
                "coverage": coverage,
                "premium" : this.L10145
        }
})
```

HTML-escaping in a helper

If a helper needs to HTML-escape a string you can call `Handlebars.escapeHTML()` in that function.
For example, if you would want to HTML-escape the value of `this.field`, but not " " or " ", you could write:
`function() { return "<p>" + Handlebars.escapeHTML(this.field) + "</p>"; }`
In the original Handlebars library this function is called `Handlebars.escapeExpression()`.

Tip

As a rule of thumb, put as much content in the .hbs file as possible and let helpers generate as little content as possible.

Partials

Partials are normal Handlebars templates that may be called directly by other templates. This topic explains how to work with Handlebars partials in OL Connect.
Information about Handlebars **templates** in OL Connect can be found in the topics "Handlebars templates" on page 291.

Note

The information in this Online Help focuses on the implementation of Handlebars in OL Connect. For general information about Handlebars and how to use it, see the following web sites: https://handlebarsjs.com/ and https://devdocs.io/handlebars.

Creating a partial

Any Handlebars template, including remote Handlebars templates, can be used as a partial in a Handlebars template in OL Connect.

To create a partial in an OL Connect template, simply create a Handlebars template; see "Creating a Handlebars template" on page 292.

Tip

To keep an overview you could group the partials in a sub folder of the Snippets folder on the Resources pane.

How to register a partial

It is not mandatory to *register* a partial, but it can be useful, for example if you want to work with dynamic partials. Registered partials can be referred to in Handlebars expressions and functions by the specified name; see "Using a registered partial in a template" on the next page.

To register a partial, use the following function in a script: Handlebars.registerPartial ('partialName', 'template') (for details see: "Handlebars API" on page 464).

For example, this line of code registers a template (clauses.hbs) as a partial with the name 'clauses':

```
Handlebars.registerPartial('clauses', 'snippets/partials/clauses.hbs');
```

Tip

Partials registered in a Control Script will be available in all standard scripts. Control Scripts are executed first. (See "Control Scripts" on page 429.)

Note

Registering multiple partials with a single call is not supported in OL Connect.

Using partials in Handlebars templates

To include a partial in a Handlebars template, insert an expression that refers to it, for example: {{> partialname.hbs}}.
Relative paths (starting with snippets/) and arbitrary URLs (starting with file:// or http:// or https://) are supported.
The file name can be:

- the name of an .hbs snippet in the template (for example:`{{> snippets/Snippet 1.hbs}}`
- the name of an .hbs snippet on disk (starting with file:///)
- the name of a remote .hbs snippet (starting with http:// or https://)

With a snippet on disk, the complete syntax is: file://<host>/<path>. If the host is "localhost", it can be omitted, resulting in **file:///<path>** - note the **three** forward slashes after **file:**.
In the remainder of the path you can either use escaped backward slashes:
"file:///C:\\Users\\Administrator\\Desktop\\Handlebars_LoadFile.hbs"
or forward slashes:
"file:///C:/Users/Administrator/Desktop/Handlebars_LoadFile.hbs"

Note

A partial is compiled automatically when the template in which it is inserted is rendered.

Using a registered partial in a template

Registered partials (see "How to register a partial" on the previous page) can be referred to by **name** instead of file name, using the following syntax: {{> partialName}}.

For example, a partial that is registered as follows:
`Handlebars.registerPartial('clauses', 'snippets/partials/clauses.hbs');`
can be included in a Handlebars template with the following expression: {{> clauses}}.

When a partial is missing

If there is no partial with the specified name, Handlebars can generate some other content, called **failover content**. Failover content can be specified using block syntax. The expression that calls the partial is the start of a block. An expression with a closing tag and the name of the partial signals the end of the block. The content between these expressions will be used if the called partial is missing.

For example: {{#> myPartial }} Failover content {{/myPartial}} will render "Failover content" if the "myPartial" partial is not registered.

Dynamic partials

To load partials dynamically, based on data, you could use a block helper (see "Using logic in a Handlebars template: helpers" on page 456 and "Using Handlebars templates: examples" on page 294).

It is also possible to dynamically select a partial by using a **sub expression** between parentheses. The sub expression should evaluate to the name of a partial.

For example, if whichPartial is a function that returns the name of a partial, `{{> (whichPartial) }}` in a Handlebars template calls `whichPartial` and then renders the partial whose name is returned by this function.

Note that to use a function in this way, it must be registered as a *helper*. See "Adding custom helpers" on page 459.

Inline partials

Inline partials are supported as well. See the two examples at https://docs.w3cub.com/handlebars/partials#inline-partials.

Unsupported features

The following Handlebars **partials** features are not supported in OL Connect.

- Passing a custom context as a parameter.
- Passing hash parameters (seehttps://devdocs.io/handlebars/partials#partial-parameters).

Handlebars API

This topic lists the functions of Handlebars that are supported in Designer scripts. It also lists features that are unsupported in OL Connect.

> **Note**
>
> The information in this Online Help focuses on the implementation of Handlebars in OL Connect. For general information about Handlebars and how to use it, see the following web sites: https://handlebarsjs.com/ and https://devdocs.io/handlebars.

Functions

compile ('template')	Compiles the specified Handlebars template* into a function that can be called with data. For example: `const template = Handlebars.compile ('snippets/Template1.hbs'); const result = template(record);` **Note**: Passing options is not supported in OL Connect.
render('template', data)	Renders a specified Handlebars template.* If data is passed, that data is used to render the template. If no data is passed, the current **record** is used to render the template.

	Handlebars.render("snippets/Template.hbs") is identical to: const compiledTemplate = Handlebars.compile("snippets/Template.hbs") compiledTemplate(record) The result of this function is HTML. See also: "Handlebars templates" on page 291.
registerPartial ('name', 'template')	Registers a template* as a partial with the specified name. For example, this line of code registers a template (clauses.hbs) as a partial with the name 'clauses': `Handlebars.registerPartial('clauses', 'snippets/partials/clauses.hbs');` The partial can then be referred to by its name in Handlebars templates as well as functions using {{> partialName}}. For example: {{> clauses}}. See: "Partials" on page 461.
registerHelper()	Registers a function as helper with the specified name. **Note**: Registering multiple helpers with a single call is not supported. See: "Handlebars expressions" on page 453.
escapeHTML()	HTML-escapes the supplied string. This means that those characters that have a special function in HTML: **& < > \" ' ` =** are replaced with HTML character references. For example: **&** will be changed into **&**. See: "HTML values" on page 455.

* In all these functions, 'template' can be:

- the name of an .hbs snippet in the template (for example:`{{> snippets/Snippet 1.hbs}}`
- the name of an .hbs snippet on disk (starting with file:///)
- the name of a remote .hbs snippet (starting with http:// or https://)
- a string that contains HTML and Handlebars expressions.

With a snippet on disk, the complete syntax is: file://<host>/<path>. If the host is "localhost", it can be omitted, resulting in **file:///<path>** - note the **three** forward slashes after **file:**.
In the remainder of the path you can either use escaped backward slashes:
"file:///C:\\Users\\Administrator\\Desktop\\Handlebars_LoadFile.hbs"
or forward slashes:
"file:///C:/Users/Administrator/Desktop/Handlebars_LoadFile.hbs"

> **Note**
>
> It is not possible to use `loadhtml()` in Handlebars functions. HTML with Handlebars expressions is not necessarily valid HTML. Processing it with an HTML parser might break both the Handlebars expressions and the HTML.

Unsupported features

The following features are not supported in Handlebars **templates** in OL Connect:

- The `SafeString` class. It is not needed since the result of a block helper is not HTML-escaped in OL Connect.
- Inline escapes.
- Raw block helpers.
- Passing options when compiling a Handlebars template.

The following Handlebars **partials** features are not supported in OL Connect.

- Registering multiple partials with a single call.
- Passing a custom context as a parameter.
- Passing hash parameters (seehttps://devdocs.io/handlebars/partials#partial-parameters).

The following Handlebars **helpers** features are not supported in OL Connect.

- The `lookup` helper and `log` block helper.
- The data variable **@key**. OL Connect only supports iterating over arrays and tables, not over arbitrary objects.
- Registering multiple helpers with a single call.
- The `blockHelperMissing` and `helperMissing` hooks.
- Passing `options.hash`, `options.helpers` and `options.partials` arguments to helpers. (`options.data`, `options.fn` and `options.inverse` are supported.)

Translating templates

OL Connect has a built-in feature to automatically generate output and previews from one template in different languages, and to pluralize certain words in the output.
When OL Connect generates output or creates a preview for a template, it looks for a translation file that matches the current "Locale" on page 336. If there is one, OL Connect uses it to localize any elements that are tagged for translation, and to pluralize words as indicated.

For this feature to work, a template needs to have:

- Content that is **tagged for translation** (see: "Tagging elements for translation" on page 469) and, optionally, pluralization (see "Pluralization" on page 472).
- One or more **translation files** (.po). Each file gives translations in one language.

The files that hold the actual translations are not made in the Designer. They can be made in any translation tool that accepts a .pot file as input, and outputs .po files.

- The **.pot file** is exported from the "Translations pane" on page 594 in the Designer. The file contains a list of texts in the base language. In OL Connect this is the language in which the template was originally written. (A template's language can be set via the "Locale Settings" on page 515 dialog.)
- A **.po file** holds translated texts for one language. Each .po file has an entry by which OL Connect recognizes the target language.

(See: "Exporting and importing translation files" on page 474.)

These file types are widely used in translation software, so you can team up with a professional translator or translation agency to translate your templates, or you can do it yourself using a free online tool, such as the Poedit software.

Other ways to translate a template

The built-in translation feature is easy to use for labels and short texts in for instance invoices, web forms, transaction email messages, and pack lists. It is less suitable for long texts like insurance policies or terms and conditions.

- For longer texts it is recommended to use a **content management system** in combination with scripts.
- It is also possible to use **snippets** or separate **sections** for translation purposes in combination with scripts.

Translating a template

To make use of the built-in translation feature, take the following steps.

1. Tag content for translation. See "Tagging elements for translation" on the next page.
2. Optional: enable the pluralization option on any translation entry that should conform with the number in a certain data field; see "Pluralization" on page 472.
3. Export the entries on the Translations pane to a **.pot** file. See "Exporting a file for translation (.pot)" on page 474.
4. Open the .pot file in a translation tool (e.g. Poedit), translate the texts and/or enter the (text with) plurals.
   Or, if you work with a translator or translation agency, send them the .pot file.
   The result is a **.po** file (one per language).
5. Import the .po file or files. See "Importing translations (.po)" on page 475.
6. Set the template's Locale, or select the field that contains a Locale value, in order to get output in the correct language. See "Changing the locale" on page 336.

Translating snippets

"Snippets" on page 285 get translated when they are inserted in the output, if the text is tagged for translation (see "Tagging text in snippets" on page 471).

Translating content that is inserted by a script

Personalization scripts may add content to the output. OL Connect will apply translations to that content at the moment it is added to the template, for example with the `html()` or `replaceWith()` function. Translations are applied if that content is marked for translation, and if there is a matching translation entry (see "Tagging text that is inserted by a script" on page 471).

Translating content via a script

If a template has been prepared for translation, personalization scripts that add content to the output may translate that content using the "translate()" on page 930 function.

Tagging elements for translation

In order to mark text for translation, you have to tag the HTML element that holds the text.

When you add elements, such as text, images or a table, to the content of a template, you are actually constructing an HTML file. It is possible to edit the source of the HTML file directly in the Designer; see "Editing HTML" on page 193.

However, to tag an element for translation, you don't have to go into the Source view.

1. Place the cursor in the text that should be translated and select the HTML element in the breadcrumbs (see "Selecting an element" on page 196) or select the element via the Outline view. Elements that contain text are paragraphs, headings, table cells, buttons, labels etc.
2. Open the Translations pane. If it isn't visible, select: **Window > Show View > Translations** from the menu.
3. Click the **Tag Element** button in the toolbar of the **Translations** pane. This adds the `data-translate` attribute to the element. (Attributes are visible on the "Source tab" on page 597.)

If there was no translation entry with the same text already, the text of the tagged element is copied to a new translation entry on the **Translations** pane.
One translation entry works for all HTML elements that are tagged for translation and have exactly the same text.

> **Note**
>
> Remember, only text in HTML elements that are tagged for translation get translated in the output.
>
> Changing the text in a tagged element will break the relationship with the translation entry. This means that the text will not be translated. Edit the entry in the Translations

pane or use the **Sync** button on the "Translations pane" on page 594 to restore the connection.

Note that placeholders (e.g. @Name@) in the template will get replaced with data before translation takes place. If a translation entry contains a placeholder, the text will then no longer match the translation entry.

Also note that, by default, *all* text in a Dynamic Table cell is replaced with data. To prevent this, use the technique described here: Mixing text and data in one cell.

Same text, different translation: add a context

Should the same text be **translated differently** in different locations? Then you need to add a 'context' to the translation entry:

1. **Double-click** the translation entry, or select an element and press the **Ctrl** key when you click the **Tag Element** button.

   Tip

   Pressing the **Ctrl** key when you click the **Tag Element** button opens the Translation String Options dialog.

2. In the **Context** field, describe where the source text is used, for example: "This text is in the head of a table", or "Table|Head"

The description is added to the <data-translate> attribute of the HTML element in which the source text is located, as its value, for example: <data-translate="Table">.
Note that a translation will only be applied if the value of the <data-translate> attribute of the respective HTML element is *exactly* the same as the given Context (case-sensitive).

Tagging text in snippets

If the text that you want to tag for translation is located in a snippet, do not tag it in the section where the snippet is used. Instead, open the snippet (see "Snippets" on page 285) and tag the elements inside the snippet as described above.

Tagging text that is inserted by a script

OL Connect will also apply translations to content that is inserted by personalization scripts, but only if that content is marked for translation, and if there is a matching translation entry.

1. On the Translations pane, click on the **New Empty String** button to create the translation entry.
2. Make sure that any element in which the text is inserted, are tagged for translation. There are several ways to do that:
    - Let the script insert the tagged HTML element as well as the text, for example: `results.html( "<p data-translate>Bill to @CustNumber@</p>" );`.
    - If the script inserts text into an existing, empty HTML element, switch to the Source view and add the `data-translate` attribute to the element (e.g. <p id="p1" data-translate>).

Data placeholders in translation entries

A translation entry may include placeholders for data fields, e.g. "Dear @name@". (For more information about placeholders see "Variable Data" on page 351.)
Placeholders must not be translated; otherwise the personalization script will no longer replace them with data

Typically, translators are familiar with entries that contain variables, but you may add a comment explaining that the placeholder is used as a variable and that it should not be translated.

To add a comment, simply double-click the entry in the **Translations** pane and enter your comment in the **Comments** field. Comments will be added to the POT file and are visible to translators in their translation tools.

HTML tags in translation entries

A translation entry may contain HTML tags, for example when part of the text is styled or when there is a hyperlink in the text. HTML tags must not be translated or removed. Normally, translators will recognize a simple bold or italic tag (... or ... respectively), but more complex elements like hyperlinks could cause problems. The translator may accidentally modify the tag, and break the hyperlink for example.

One way to avoid this is to split the text around and inside HTML tags into separate chunks, no matter how short. Select the respective text in the template and select **Wrap in Span** from the contextual menu. Subsequently tag the new element for translation.

Alternatively you could also add a comment explaining that the HTML tag should not be changed.

To add a comment, simply double-click the entry in the **Translations** pane and enter your comment in the **Comments** field. Comments will be added to the POT file and are visible to translators in their translation tools.

Pluralization

Ideally, in templates with variable data, the text automatically adjusts to numbers in the data. Take for example the following line in a template:
"This order contains @num_products@ product."
In the output, the text "@num_products@" will be replaced with the value of a data field, a number in this case. (See "Variable Data" on page 351.)
The word "product" should only remain the same when the actual number of products is 1:
"This order contains 1 **product**".
In all other cases it should be changed into "products", e.g.:
"This order contains 10 **products**".

This topic explains how to make OL Connect dynamically select a singular or plural word, depending on the value of a data field.

Adding plurals

Entering plurals, or text with plurals, is a natural part of the translation process of a template. This process starts by creating translation entries as described in the following topic: "Translating templates" on page 467.

Note that it isn't required to translate the template effectively. You may also create a translation file for the original language of a template and use it just for pluralization by entering plurals only.

To enable pluralization on a translation entry:

1. Double-click the respective entry on the **Translations** pane.
2. Check the **Pluralization** option.
3. Select the data field that holds the number.

> **Note**
>
> Pluralization cannot be based on a data field in a detail table.

4. Now there are two possibilities:
   - Click OK. Proceed with the translation process after you've created all translation entries and enabled pluralization as needed. For further instructions see "Translating templates" on page 467.
   - Enter the singular and plural and click OK. The entries are stored in the template itself.
     OL Connect will automatically fall back to these entries when there is no translation file (.po) that matches the "Locale" on page 336. The singular is used when the value of the data field is 1, and the plural is used in all other cases.

> **Note**
>
> In some languages, like Arabic and Polish, words have multiple plurals for use with different numbers. Multiple plurals can only be entered in a translation tool. See "Translating templates" on page 467.

Exporting and importing translation files

In the process of the translation of a template, two types of translation files are involved: **Portable Object (.po)** files and a **Portable Object Template (.pot)** file. These are standardized file types following the gettext format.

This topic explains how to export the .pot file and how to import a .po file.

Tip

Translation tools are capable of **combining** .pot files. Combining .pot files may also result in .po files that contain translations for entries in multiple templates. This can be an interesting feature, especially when there are multiple templates in a project, as this eliminates the need to maintain individual .po files for each template.

For information about the translation process, see "Translating templates" on page 467.

Exporting a file for translation (.pot)

A .pot file holds a list of texts in the base language. In OL Connect this is the language in which the template was originally written. The file contains the entries in the Translation pane, including any comments.

Once all translatable elements in a template are tagged (see "Tagging elements for translation" on page 469), you can create the .pot file:

1. Click the **Export** button on the toolbar of the **Translations** pane.
2. Specify the name and select a folder for the .pot file.

Importing translations (.po)

A .po file holds translated texts for one language. PO files can be imported into the template file or used remotely.

Importing a .po file

Importing a .po file file is easy: **drag & drop** or **copy & paste** the file from the file system to the **Translations** folder in the **Resources** pane.
Alternatively, right-click the Translations folder in the Resources pane and select **New Translation > PO files...**

Editing an imported .po file

When you double-click a .po file in the Translations folder it will be opened in the text editor of the Designer, where you can read and edit it. If you are working with a translator or translating agency, remember to send them a copy of the modified .po file to keep the .po files in sync.

Updating an imported .po file

OL Connect Designer monitors the .po file on disk (in the folder from which it was last imported) for changes. When a .po file on disk has changed, the respective .po file in the template gets a small asterisk behind its name. Right-click the file in the Translations folder and select **Update** from the contextual menu to import the latest changes.

Using a remote .po file

In order to use a remote .po file, you must add the URL to the Translations folder on the Resources pane.

1. Right-click the **Translations** folder on the **Resources** pane, and select **New Translation > New Remote PO file**.
2. Enter a name for the file as it appears in the Translations folder. For better management, it's best to use the same file name as the remote resource.

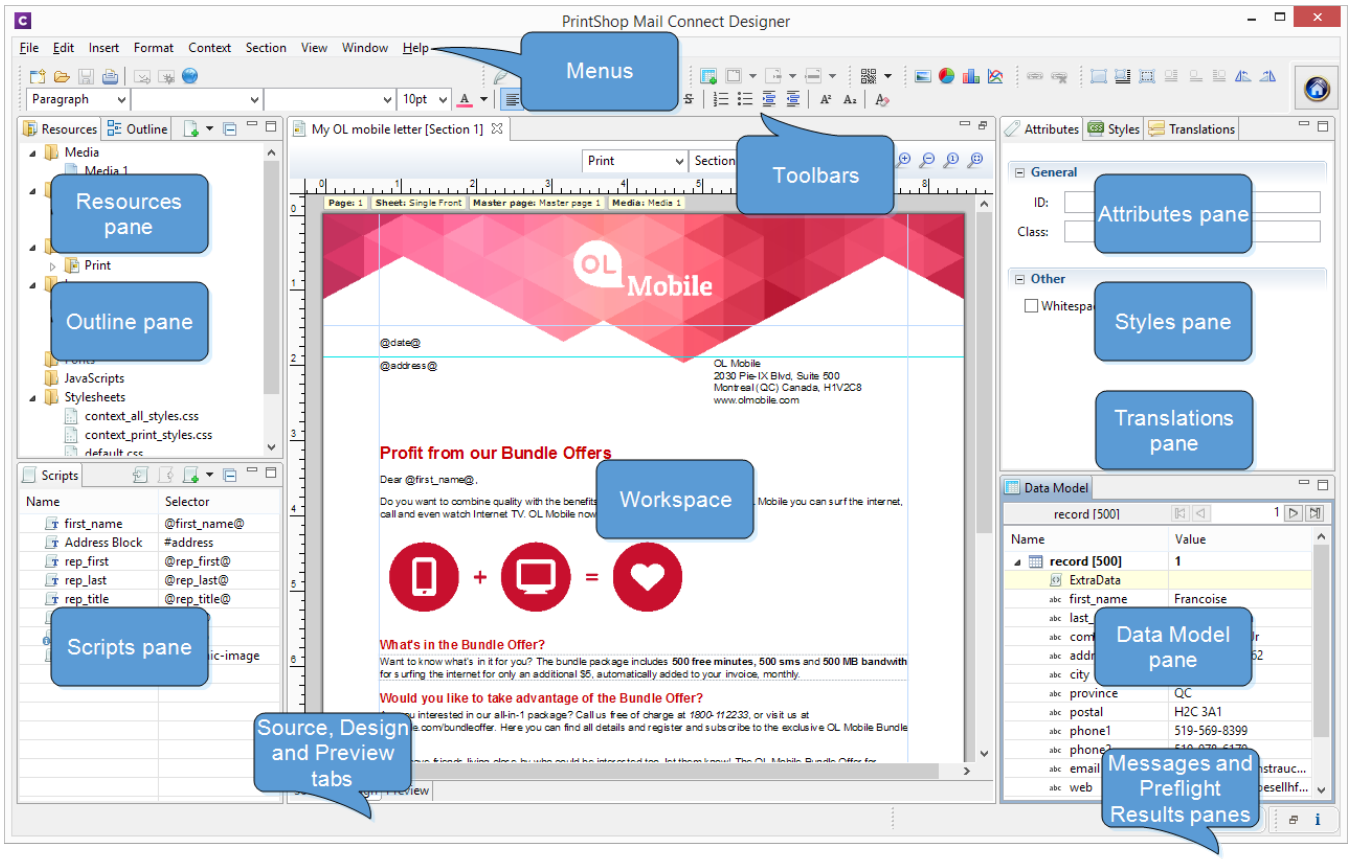
   Enter the **URL** for the remote .po file.
   If the file is located on an external web server, this must be a full URL, including the http:// or https:// prefix, domain name, path and file name.
   If the file is located on your network you can click the Browse button or enter the path and file name manually. The complete syntax with the "file" protocol is:
3. `file://<host>/<path>`
   . If the host is "localhost", it can be omitted, resulting in
   `file:///<path>`
   , for example: file:///c:/resources/translations/de-DE.po.
   If the file is located on another server in your network, the path must contain five slashes after "file:".

Designer User Interface

The main ingredients in the Designer's user interface are the following:

- "Menus" on page 568
- "Toolbars" on page 598
- "Resources pane" on page 583
- "Outline pane" on page 583
- "Attributes pane" on page 577
- "Styles pane" on page 593

- "Translations pane" on page 594
- Parameters pane (see Runtime parameters)
- "Workspace" on page 595 (Source, Design and Preview tabs)
- "Data Model pane" on page 579
- "Scripts pane" on page 590
- "Preflight Results and Messages" on page 581



Dialogs

Dialogs can allow you to perform a command or make settings. They can also ask you a question or provide you with information or progress feedback.

Here is a list of all dialogs:

Attachments dialog

The Attachments dialog lets you add and remove static Email attachments. For more information about email attachments, see: "Email attachments" on page 187.

- **Add**: Click this button to open the Select File dialog (see below).
- **Delete**: Deletes the file from the list of attachments. It doesn't remove any files from the template's resources.
- **Attach Print context as PDF**: If a Print Context exists in the template, its output will be generated and a PDF version of it will be attached to the outgoing email. This option can be overridden by the setting in the Send Test Email or Send Email dialog.

Select File dialog

Resources: lists the images that are present in the Images folder on the Resources pane. A preview of the selected image will be shown at the right.

Disk: lets you select a file that resides in a folder on a hard drive that is accessible from your computer. If possible, a preview of the selected file will be shown.

- **Path**. The complete syntax of the path is: file://<host>/<path>. Note: if the host is "localhost", it can be omitted, resulting in file:///<path>, for example: file:///c:/documents/mydocument.doc.
- **Browse**: opens an explorer window to browse folders and select a file.

Url: lets you select a file from a specific web address. Select the protocol and then enter the URL (for example, http://www.mysite.com/zipfiles/myfile.zip). If the selected file is an image, a preview will be shown below.

- **Protocol**: **http** or **https**.

> **Note**
>
> Certain characters are invalid in a URL (for example, '$', '%', and '&') and must be percent-encoded. The same applies to a file path, since that actually is a URL that starts with the file protocol.

Note that even a space character is invalid in a URL. Spaces in a URL are supported for backward compatibility, but it is recommended to percent-encode a space character as %20.

The option **Save with template** is available when choosing an image from disk or by URL. This inserts the image in the **Images** folder on the **Resources** pane. If not saved with the template, the image will remain external.

Note

External files need to be available when the template is merged with a record set to generate output, and their location should be accessible from the machine on which the template's output is produced. External files are updated (retrieved) at the time the output is generated.

Bar Chart Properties dialog

The Bar Chart dialog appears when a Bar Chart object is right-clicked and the **Chart...** option is clicked. It determines how the Bar Chart is displayed in output and in Preview mode (see "Business graphics" on page 260).

All of the settings in this dialog are equivalent to a property in the amCharts library (see: amCharts' Class Reference). The description of each setting specifies the corresponding property. When you've made a setting, you can also find the property on the Source tab.

General tab

Most settings on the General tab correspond to properties of the AmSerialChart class in the amCharts library; see: AmSerialChart.

- **General Group**:

  Display grid above graph: Check to display the grid on top of the bars so that it is always visible. (Equivalent to the
- `gridAboveGraphs`
  property; see: gridAboveGraphs.)

Rotate: Check to rotate the chart 90 degrees so that the bars are horizontal starting from the left. (Equivalent to the
- `rotate`
property; see: rotate.)

Stack Series: Check to stack the values in one bar, instead of having one bar per value. (Equivalent to the
- `stackType`
property of the
`ValueAxis`
; see: stackType.)

- **Text Group**: Determines how text is displayed in labels and legends.

  Font: Enter the font-face to use to display text. The font must be installed on the system and defaults to Verdana if the font is not found. (Equivalent to the
  - `fontFamily`
  property; see: fontFamily.)

  Size: Enter the size of the font. Defaults to 11. (Equivalent to the
  - `fontSize`
  property; see: fontSize.)

  Color: Select the color in which to display text: click the colored square to open the Color Picker dialog ("Color Picker" on page 492), or enter a valid hexademical color (HTML Hex Color) or a predefined CSS color (CSS color names). (Equivalent to the
  - `color`
  property; see: color.)

- **3D Group**: Creates a 3D effect if both settings in this group are higher than 0.

  Depth: The depth of the 3D part of the plot area. (Equivalent to the
  - `depth3D`
  property; see: depth3D.)

  Angle: The angle of the 3D part of the plot area. (Equivalent to the
  - `angle`
  property; see: angle.)

Value Axis tab

Settings on the Value Axis tab are equivalent to properties of the ValueAxis class in the amCharts library; see: ValueAxis.

- **Title group**:

  Label: Enter a label for the value axis (the Y axis, or the X axis if the graph is rotated). (Equivalent to the
  - `title`
  property; see: title.)

  Bold: Check this option if the title should be bold. (Equivalent to the
  - `titleBold`
  property; see: titleBold.)

  Color: The default label color is black. To select a custom color for the label, click the colored square to open the Color Picker dialog ("Color Picker" on page 492), or enter a valid hexademical color (HTML Hex Color) or a predefined CSS color (CSS color names). (Equivalent to the
  - `titleColor`
  property; see: titleColor.)

  Font Size: Enter a font size for the label. (Equivalent to the
  - `titleFontSize`
  property; see: titleFontSize.)

- **Grid group**:

  Color: Select a color for the grid lines that divide the value axis: click the colored square to open the Color Picker dialog ("Color Picker" on page 492), or enter a valid hexademical color (HTML Hex Color) or a predefined CSS color (CSS color names). (Equivalent to the
  - `gridColor`
  property; see: gridColor.)

  Opacity: Enter the opacity percentage of the grid. Default is 15%. 100 is fully opaque, 0 is transparent. (Equivalent to the
  - `gridAlpha`
  property; see: gridAlpha.)

  Thickness: Enter a thickness for the grid lines. Default is 1. (Equivalent to the
  - `gridThickness`
  property; see: gridThickness.)

  Tick Length: Length of the tick marks. (Equivalent to the
  - `tickLength`
  property; see: tickLength.)

- **Axis group**:

Color: Select a color for the value axis: click the colored square to open the Color Picker dialog ("Color Picker" on page 492), or enter a valid hexademical color (HTML Hex Color) or a predefined CSS color (CSS color names). (Equivalent to the

- `axisColor` property; see: axisColor.)

Opacity: Enter the opacity in percentage for the axis. 100 is fully opaque, 0 is transparent. Set opacity to 0 to hide the axis. (Equivalent to the

- `axisAlpha` property of the `ValueAxis`; see: axisAlpha.)

Thickness: Enter the thickness of the axis. (Equivalent to the

- `axisThickness` property of the `ValueAxis`; see: axisThickness.)

Labels: Check this option to make the labels on the value axis visible. This is often useful in rotated charts. (Equivalent to the

- `labelsEnabled` property of the `ValueAxis`; see: labelsEnabled.)

Frequency: Defines per how many grid lines a label will be placed. (Equivalent to the

- `labelFrequency` property of the `ValueAxis`; see: labelFrequency.)

Category Axis tab

Settings on the Category Axis tab correspond to properties of the CategoryAxis class in the amCharts library; see: CategoryAxis.

- **Title group**:

  Label: Enter a label for the category axis (the X axis, or the Y axis if the graph is rotated).(Equivalent to the

  - `title` property; see: title.)

Bold: Check if you want the title to be bold. (Equivalent to the
- `titleBold`
property; see: titleBold.)

Color: Select a custom color for the label (default is black): click the colored square to open the Color Picker dialog ("Color Picker" on page 492), or enter a valid hexademical color (HTML Hex Color) or a predefined CSS color (CSS color names). (Equivalent to the
- `titleColor`
property; see: titleColor.)

Font Size: Enter a font size for the title. (Equivalent to the
- `titleFontSize`
property; see: titleFontSize.)

- **Grid group**:

  Color: Select a color for the grid lines that divide the category axis: click the colored square to open the Color Picker dialog ("Color Picker" on page 492), or enter a valid hexademical color (HTML Hex Color) or a predefined CSS color (CSS color names). (Equivalent to the
  - `gridColor`
  property; see: gridColor.)

  Opacity: Enter the opacity percentage of the grid. Default is 15%. 100 is fully opaque, 0 is transparent. (Equivalent to the
  - `gridAlpha`
  property; see: gridAlpha.)

  Thickness: Enter a thickness for the grid lines. Default is 1. (Equivalent to the
  - `gridThickness`
  property; see: gridThickness.)

  Position: Specifies if a grid line is placed on the centre of a cell or on the beginning of a cell. (Equivalent to the
  - `gridPosition`
  property; see: gridPosition.)

> **Tip**
>
> The position of the ticks (which is Middle by default) does not move with the grid. To change the tick position, add the following to the `categoryAxis`

section on the Source tab:
`"tickPosition": "start"`

.

- **Tick Length**: Length of the tick marks. (Equivalent to the `tickLength` property; see: tickLength.)

> **Tip**
>
> By default, one label per grid line will appear on the Category axis. To change that frequency, add the following to the `categoryAxis` section on the Source tab:
> `"labelFrequency": "2"`
>
> (replace 2 by the desired number of grid lines).

- **Axis group**:
  - **Color**: Select a color for the value axis: click the colored square to open the Color Picker dialog ("Color Picker" on page 492), or enter a valid hexademical color (HTML Hex Color) or a predefined CSS color (CSS color names). (Equivalent to the `axisColor` property; see: axisColor.)
  - **Opacity**: Enter the opacity in percentage for the axis. 100 is fully opaque, 0 is transparent. Set opacity to 0 to hide the axis. (Equivalent to the `axisAlpha` property; see: axisAlpha.)
  - **Thickness**: Enter the thickness of the axis. (Equivalent to the `axisThickness` property; see: axisThickness.)
  - **Auto Wrap**: Specifies if axis labels (when horizontal) should be wrapped if they don't fit in the allocated space. (Equivalent to the `autoWrap`

property; see: autoWrap.)

Graphs tab

Most settings on the Graphs tab correspond to properties of the AmGraph class in the amCharts library; see: AmGraph.

- **Labels**:

  Text: Text of the data labels. You can use tags like [[value]], [[description]], [[percents]], or[[category]]. (Equivalent to the
  - `labelText`
  property; see: labelText.)

  Position: Position of the data labels. Possible values are: "bottom", "top", "right", "left", "inside", "middle". (Equivalent to the
  - `labelPosition`
  property; see: labelPosition.)

  Rotation: Rotation of the data labels. Supported notations:
  - `0.0`
  ,
  `0.0°`
  or
  `0.0 deg`
  . (Equivalent to the
  `labelRotation`
  property; see: labelRotation.)

  Offset: Vertical offset of the data labels. A positive offset moves the data labels up. A negative value moves them down. (Equivalent to the
  - `labelOffset`
  property; see: labelOffset.)

  Anchor: Select whether the start, middle or end of a data label should be anchored to the bar. This setting moves the labels horizontally. (Equivalent to the
  - `labelAnchor`
  property; see: labelAnchor.)

- **Columns**:

  Width: Specify a bar width between 0 and 1. (This is a relative width, not pixel width). (Equivalent to the
  - `columnWidth`
  property; see: columnWidth.)

Spacing: The gap in pixels between two columns of the same category. (Equivalent to the

- `columnSpacing` property of the `AmSerialChart` class; see: columnSpacing.)

Legend tab

Settings on the Legend tab correspond to properties of the AmLegend class in the amCharts library; see: AmLegend.

Show Legend: Specifies if legend is enabled or not. (Equivalent to the

- `enabled` property; see: enabled.)

- **Legend Group**: Defines how the legends are shown.

  Equal label widths: Check so that all labels are of equal width in the Legends box. The Legend's width will accommodate the largest value. (Equivalent to the

  - `equalWidths` property; see: equalWidths.)

  Position: Use the drop-down to select where the legend is shown: at the Right, Left, Top or Bottom. (Equivalent to the

  - `position` property; see: position.)

  Align: Use the drop-down to select how to align the text in the labels: Left, Middle or Right. (Equivalent to the

  - `align` property; see: align.)

  Horizontal Space: Horizontal space between legend items, in pixels. (Equivalent to the

  - `spacing` property; see: spacing.)

  Vertical Space: Enter a numerical value (in pixels) to define the vertical space between legend items, and also between the legend border and the first and last legend item. (Equivalent to the

  - `verticalGap` property; see: verticalGap.)

  - **Max Columns**: Enter a numerical value to define the maximum number of columns in the legend. If the Legend's position is set to "right" or "left", this is automatically

set to 1. (Equivalent to the
`maxColumns`
property; see: maxColumns.)

- **Labels Group**: Defines if and how labels are shown in the Legend.

  Text: Enter the text used to display the labels;
  - `[[title]]`
  is a variable that will be replaced with the title of the graph. (Equivalent to the
  `labelText`
  property; see: labelText.)

- **Markers Group**: Defines how the Legend's Markers look. Markers are icons with a color matching the Legend with its corresponding line.

  Type: Use the drop-down to select in which shape the Markers are displayed; "none" hides the Markers completely. (Equivalent to the
  - `markerType`
  property; see: markerType.)

  Size: Enter the size (in pixels) for the Markers to be displayed. (Equivalent to the
  - `markerSize`
  property; see: markerSize.)

  Label Gap: Enter the distance (in pixels) between the legend marker and legend text. (Equivalent to the
  - `markerLabelGap`
  property; see: markerLabelGap.)

  Border Width: Use the drop-down to define the thickness of the border added to the Markers. The default value (0) means the line will be a "hairline" (1 px). In case the Marker type is line, this style will be used for the line thickness. (Equivalent to the
  - `markerBorderThickness`
  property; see: markerBorderThickness.)

  Border Color: Color of the Legend's border. Enter a valid hexademical color (HTML Hex Color) or a predefined CSS color (CSS color names), or click the colored square to open the Color Picker dialog ("Color Picker" on page 492). (Equivalent to the
  - `markerBorderColor`
  property; see: markerBorderColor.)

  Border Opacity: Enter a numerical value between 0 and 100 to define the opacity (in percentage) of the border. (Equivalent to the
  - `markerBorderAlpha`

property; see: markerBorderAlpha. When specified on the Source tab, the value should be between 0 and 1, e.g. 0.8.)

Source tab

The JSON on the Source tab reflects the choices made in the other tabs and, more importantly, provides the possibility to add in any amCharts configuration option that is unavailable via the other tab menus. For more information see: "Adding and editing properties manually" on page 265.

Box Formatting dialog

The Box Formatting dialog is accessible by clicking inside a box in the template and then selecting **Format > Box** in the menu.

All settings in this dialog are in fact CSS properties. Cascading Style Sheets (CSS) were originally designed for use with web pages: HTML files. Since Designer templates are HTML files, they are styled with CSS. To learn how to use CSS in the Designer, see "Styling and formatting" on page 294 and "Styling templates with CSS files" on page 296. For information about specific properties and their options, see W3Schools CSS Reference.

Note

When no unit is added to a geometry value, such as the width, height, top or margin, the default unit will be added to the value; see "Print preferences" on page 390.

Note

Using viewport based units (vw, vh, vmin, vmax) in Print sections is not recommended. This may cause differences between the preview in the Designer and the printed output.

For information about the different types of Boxes, see "Boxes" on page 256.

Box tab

- **General group**:
  - **Width**: Set the width of the box in measure or percentage. When no unit is entered, the default unit will be added to the value (see "Print preferences" on page 390). Equivalent to the CSS `width` property.
  - **Height**: Set the height of the box in measure or percentage. When no unit is entered, the default unit will be added to the value (see "Print preferences" on page 390). Equivalent to the CSS `height` property.
  - **Angle**: Set the rotation angle of the box in clockwise degrees. Equivalent to the CSS `transform:rotate` property.
  - **Corner radius**: Set the radius of rounded border corners in measure or percentage. Equivalent to the CSS `border-radius` property.
  - **Display**: Use the drop-down or type in the value for how to display the box. Equivalent to the CSS `display` property.
  - **Overflow**: Use the drop-down or type in the value for how to handle overflow (text that does not fit in the current size of the box). Equivalent to the `overflow` property.
- **Text wrap group**:
  - **Float**: Use the drop-down or type in the value for how to float the box, if the box is not in an absolute position (see Position, below). Equivalent to the CSS `float` property.
  - **Clear**: Use the drop-down or type in the value for clearing pre-existing alignments. Equivalent to the CSS `clear` property.
- **Positioning**: Note that it depends on both the Context and the type of Box, which settings are available. For information about the different types of Boxes, see "Boxes" on page 256.
  - **Position**: Use the drop-down or type in the value for the type of positioning for the box. Equivalent to the CSS `position` property (see "Using the CSS position property" on page 314).
  - **Top** (Web Context): Set the vertical offset between this box and its parent's top position. Equivalent to the CSS `top` property.
  - **Left** (Web Context): Set the horizontal offset between this box and its parent's left position. Equivalent to the CSS `left` property.
  - **Y-offset** (Print Context): Sets the vertical offset between this box and the current page's top. This only works for Positioned boxes.

- **X-offset** (Print Context): Sets the vertical offset between this box and the current page's left edge. This only works for Positioned boxes.
- **Bottom**: Set the vertical offset between this box and its parent's bottom position. Note that you can't set the Height *and* Top (or Y-offset) *and* Bottom of a Box. Remove the Height before setting the Bottom property. Equivalent to the CSS `bottom` property.
- **Right**: Set the horizontal offset between this box and its parent's left position. Note that you can't set the Width *and* Left (or X-offset) *and* Right of a Box. Remove the Width before setting the Right property. Equivalent to the CSS `right` property.
- **Z-index**: Set the z-index of the box. The z-index defines in which order elements appear. Equivalent to the CSS `z-index` property.

Background tab

For information about backgrounds see "Background color and/or image" on page 323.

- **General group**:
  - **Color**: Specify the color of the box background: select a named color (defined in the "Colors Properties" on page 492) from the drop-down, or click the colored square to open the Color Picker dialog ("Color Picker" on page 492). Alternatively you could type a name or value in the Color field directly. It must be a predefined CSS color name (CSS color names), a hexadecimal color code (HTML Hex Color), an RGB color value, for example `rgb(216,255,170)` or a CMYK color value, for example `cmyk(15%, 0%, 33%, 0%)`.
    Equivalent to the CSS `background-color` property.
- **Background image group**:
  - **Source**: click the **Select Image** button to select an image via the "Select Image dialog" on page 545. Equivalent to the CSS `background` property.
  - **Size**: select `auto`, `cover` or `contain` (for an explanation see https://www.w3schools.com/cssref/css3_pr_background-size.asp), or type the width and height of the image in a measure (e.g. `80px 60px`) or as a percentage of the box size (e.g. `50% 50%`). Equivalent to the CSS `background-size` property.
  - **Position**: select the position for the background-image. Equivalent to the CSS `background-position` property.

Spacing tab

For information about spacing see "Spacing" on page 337.

- **Padding group**: Defines padding (spacing inside the element) in measure or percentage:
    - **All sides**: Check to set all padding to use the Top value. Equivalent to the CSS `padding` property.
    - **Top, Left, Bottom, Right**: Set padding for each side. Equivalent to the CSS `padding-left`, `padding-top`, `padding-right` and `padding-bottom` properties.
- **Margin group**: Defines margins (spacing outside the element) in measure or percentage:
    - **All sides**: Check to set all margins to use the Top value. Equivalent to the CSS `margin` property.
    - **Top, Left, Bottom, Right**: Set the margin for each side. Equivalent to the CSS `margin-left`, `margin-top`, `margin-right` and `margin-bottom` properties.

Border tab

For information about borders see "Border" on page 325.

- **Same for all sides**: Defines the border properties for all sides using the Top properties. Equivalent to the CSS `border` property.
- **Top, Left, Bottom, Right**: Each group defines the following properties:
    - **Width**: Specify the thickness of the border. Equivalent to the CSS `border-width`property.
    - **Style**: Specify the style of the border such as `solid`, `dashed` or `dotted`. Equivalent to the CSS `border-style` property.
    - **Color**: Specify the color of the border: select a named color (defined in the "Colors Properties" on the facing page) from the drop-down, or click the colored square to open the Color Picker dialog ("Color Picker" on the facing page). Alternatively you could type a name or value in the Color field directly. It must be a predefined CSS color name (CSS color names), a hexadecimal color code (HTML Hex Color), an RGB color value, for example `rgb(216,255,170)` or a CMYK color value, for example `cmyk(15%, 0%, 33%, 0%)`.Equivalent to the CSS `border-color` property.

Content tab

- **Copy Fit**: Check this option to automatically scale text inside the Box to the available space; see "Copy Fit" on page 311 .

- **Minimum font size**: Specify the minimum font size using a valid font measurement unit (pt, px, in, cm or mm). Do not put a space between the number and the unit. The smallest possible size is 1pt. The default is 4pt.
  When left **blank**, the font size in Design view becomes the minimum font size. Text can only be made bigger than its initial size.
- **Maximum font size**: Specify the maximum font size using a valid font measurement unit (pt, px, in, cm or mm). Do not put a space between the number and the unit. The biggest possible size is 1048pt. The default is 48pt.
  When left **blank**, the font size in Design view becomes the maximum font size. Text can only be made smaller than its initial size.
- **Fit to width only**: When this option is checked, no line breaks will be added to the text.
- **Child** (optional): When specified, the Copy Fit feature will only be applied to the given child element (an element inside the Box or Div). Specify the element by giving its ID, for example: **#product**, or class, for example: **.product** - note the dot.

Color Picker

The Color Picker dialog appears when creating a color in the formatting dialogs of certain elements, for example border colors in boxes and paragraphs. For information about how to define and apply colors and how to keep them consistent in different output channels, see "Colors" on page 328.

The dialog consists of two main parts. On the left is the color wheel that can be used to select a color **hue** by clicking anywhere on that wheel. To the right of the color wheel there is a vertical bar used to select the color **saturation**. At the top-right, two colors are shown: the **New** box displays the currently selected color, while the **Original** shows the color currently attributed to the element.

The rest of the dialog has various options for choosing colors:

- **Color Mode**: Use the drop-down to select whether the color is set as **RGB**, **CMYK** or **HEX**. The color mode determines how the color is saved in the formatting properties, and how they are printed or outputted; see "Colors" on page 328.
- **RGB group**: Enter the Red, Green and Blue color values from 0 to 255.
- **HEX**: Enter a valid HTML Hex Color.
- The **eye dropper** lets you select a color from anywhere on your desktop. To open it, click the eye dropper button next to the HEX color field.

- **CMYK group** (suitable for Print output only): Enter the Cyan, Magenta, Yellow and Black color values from 0 to 100 percent.

> **Note**
>
> Whenever one value within this dialog is changed, all the other values are adjusted to its equivalent.

Colors Properties

The Colors Properties defines and sets named colors used in the template; see "Colors" on page 328. Named colors can be used throughout the templates, in all contexts. They are visible in color selector dialogs and useable with their names in style sheets; see "Styling and formatting" on page 294.

- **Color Type Selector**: Click and use the drop-down to display which color types to show in the list: All, RGB, CMYK or Spot colors.
- **Color List**: Displays the colors, filtered using the Color Type Selector.
- **New**: Create a new color using the **Edit Color** dialog (see below and see "Colors" on page 328).
- **Edit**: Edit the currently selected color using the **Edit Color** dialog (see below and see "Colors" on page 328).
- **Delete**: Delete the currently selected color.
- **Duplicate**: Duplicate the currently selected color using the name [color]CopyX.

Edit color

You can edit the following color properties.

- **Name**: Enter the name of the color. This name should not contain spaces or special characters.
- **Create name based on values**: Check so that the name is automatically based on the color slider values below.
- **Type**: Use the drop-down to specify which type of color this should be: either a Tint or a Color.
- **Option group**: contains the options for the chosen type. Options change depending on the selected type.

- **Color**:
  - **Model**: This can be either CMYK or RGB.
  - **Spot Color**: Check to set the color as Spot Color. When Spot Colors are used, the Name must match that of the spot color used in the printer.
  - **Cyan/Magenta/Yellow/Black** (CMYK): Each slider sets a percentage for the color. Set the values using the sliders, or type in the percentage directly in the input boxes.
  - **Red/Green/Blue** (RGB): Each slider sets the values of 0-255 for the color. Set the value using the sliders or type in the value directly in the input boxes.
  - **Color Preview**: Box displaying the preview of the color (converted to RGB when relevant).
- **Tint**:
  - **Source**: Select an existing *Color* in the template. The *tint* or *opacity* will be applied to this color.
  - **Tint/Opacity**: The slider sets the percentage of tint or opacity. Set the value using the slider, or type the percentage directly in the input box.
  - **Use Opacity**: Check to set the Tint slider to use Opacity instead.

Color Settings

Color Management can keep colors consistent across different outputs by using Color Profiles. When producing output to a new device, color adjustments are made to present the color as accurately as possible on this new device.

- **Enable Color Management**: Check to disable color management and ignore embedded color profiles when importing images (with the exception of imported PDF files as it might contain a multiple tagged sub images).
- **Working Space Group**: Defines the color profiles for the current template.
  - **RGB**: Use the drop-down to select a color profile for RGB colors. The list displays ICC profiles located in "%USERPROFILE%\Connect\color-profiles\rgb".
  - **CMYK**: Use the drop-down to select a color profile for CMYK colors. The list displays ICC profiles located in "%USERPROFILE%\Connect\color-profiles\cmyk"
  - **Gray**: Use the drop-down to select a color profile for Grayscale. The list displays ICC profiles located in "%USERPROFILE%\Connect\color-profiles\gray"
- **Untagged Images Group**: Defines color profiles for any image that does not specifically have color profiles or color settings enabled.

- **RGB**: Use the drop-down to select a color profile for RGB colors. The list displays ICC profiles located in "%USERPROFILE%\Connect\color-profiles\rgb".
- **CMYK**: Use the drop-down to select a color profile for CMYK colors. The list displays ICC profiles located in "%USERPROFILE%\Connect\color-profiles\cmyk"
- **Gray**: Use the drop-down to select a color profile for Grayscale. The list displays ICC profiles located in "%USERPROFILE%\Connect\color-profiles\gray"

- **Options Group**:
  - **Rendering intent**: Use the drop-down to specify how colors are converted that are out of range of a profile. For example, you may use tricks like reducing the saturation of the entire print so that a color that is out of range still appears a bit more vibrant than ones that are in range. Rendering intents use different methods to trick the eye into believing that the print can reproduce irreproducible colors.

Context Properties dialogs

The respective Context Properties dialogs are opened via the contextual menu:

- Right-click a **context** in the Resources pane and choose one of its properties.

Which properties are available depends on the context. E.g. for a Print context, you can select either Color Output or Finishing.

Email context properties

For information about the Email context, see: "Email context" on page 174.

PDF Attachments tab

For an explanation of the options on the PDF Attachments tab, see "PDF Attachments dialog" on page 523.

Print context properties

For information about the Print context, see: "Print context" on page 132.

Finishing tab

For an explanation of the options on the Finishing tab and when they are applied, see: "Setting the binding style for the Print context" on page 135.

Color Output tab

For an explanation of the options on the Color Output tab see "Overprint and black overprint" on page 136.

Edit Label Properties

The Edit Label Properties defines how a Pie Chart Label displays its title and data. It contains two options:

- **Label**: Enter a title for Labels and Legends when they are shown (see "Pie Chart Properties dialog" on page 523).
- **Value**: Use the drop-down to select which Value to use as data within the Pie Chart as well as for Label and Legend values.

Export Template Report wizard

The Export Template Report wizard is opened through the menu: **File > Export report**.

It contains the following options:

- **File Name**: Enter the desired file name for the template report.
- **Format**: Select PDF or XML.

Click **Next** to go to the next set of options:

- **Generate Thumbnails**: The generated thumbnail images will be included in the PDF file. When exporting an XML report, the thumbnails will be saved in the same folder as the XML, in a subfolder named Thumbnails.
  - **Sections (first page)**: Save the first page of each section as a thumbnail image.
  - **Image resources**: Save all image resources as thumbnail images.
- **PDF Options**:
  - **Standard**: The standard PDF report contains a summary of the template with an overview of all the settings and resources that are used in the template: media, master pages, contexts, sections, images, scripts, runtime parameters, etc. The file properties are included as well (see "File Properties dialog" below). The report has a standard layout.

- **Custom**: To create a custom PDF report, you need two files:
  - **Template**: A template design with the desired layout and variable data. This .OL-TEMPLATE file has to be made in the Designer.
  - **DataMapper**: A data mapping configuration that provides the variable data. This .OL-datamapper file has to be made in the DataMapper module, using the standard XML template report as data sample.

> **Note**
>
> The DataMapper is included only in PlanetPress Connect and PReS Connect.

File Properties dialog

The **File Properties** dialog shows the characteristics of the template, such as the file name, author, creation date and keywords. It is also possible to add custom properties; customer name, for example.
To open the File Properties dialog, select **File > Properties**.

When a template report is generated, the file properties are part of it (see "Exporting a template report" on page 113).
The properties are available in scripts via the `template` object; see the Script API: "template" on page 960.

Description tab

- **File** (read-only): The name of the file.
- **Author**: The name of the author.
- **Company**: The company the author works for.
- **Description**: Enter a description of the template.
- **Keywords**: Enter one keyword or a list of keywords, separated by semicolons.
- **Created** (read-only): The date on which the template was created.
- **Modified** (read-only): The date on which the template was last modified.

- **Application** (read-only): the name and version of the application in which the template was last modified. The Connect version history of a template will be saved in the template file and exposed in a template report (see "Exporting a template report" on page 113).

Custom tab

On this tab custom properties can be defined. Click the first empty line, or click the Add icon, and enter a name for the property and a value.
Use the arrow icons to reorder the list of custom properties.
Custom property lists can be exported to, and imported from, a .CSV file, using the Import and Export buttons.

Find/Replace Dialog

The Find/Replace dialog can replace text within the current template. The scope of the replacement depends on the currently selected tab in the Workspace. If the Source tab is selected, the replace will affect the HTML source code. If the Design tab is selected, the replace will affect the text on the page. If the Preview tab is selected, the Replace feature is inactive.

> **Note**
>
> When replacing text in the Design tab, formatting in the replaced text will be removed. If formatting is necessary in the new text, select the Source tab before opening the Find/Replace dialog and include the required HTML tags in the replacement text.

Here are the options available in this dialog

- **Find**: The source string to find.
- **Replace with**: The string to replace the source with.
- **Direction**
  - **Forward**: Look forward from the current position of the pointer in the template or source.
  - **Backward**: Look backward from the current position of the pointer in the template or source.
- **Scope**
  - **All**: Searches in the complete text of the template or source.
  - **Selected lines**: Searches in the currently selected text or source.

- **Options**
  - **Case sensitive**: Use a case sensitive search, which differentiates **TEXT** from **text** or **TexT**.
  - **Wrap search**: Loop back from the end of the template or selection to its beginning, when the Search is at the end of the template or the selection.
  - **Whole word**: Searches for the source string as a whole word.
  - **Incremental**: With this option selected, each letter you type in the Find field causes the editor focus to move to the first complete occurrence of the text you are typing.
  - **Regular expressions**: Enables regular expressions for a search in the **Source** view of the workspace. After checking this option, you can type Ctrl + Space in either text box to view a list of regular expressions.

    > **Tip**
    >
    > The Find/Replace dialog can fill in regular expressions in the Find field by itself. Open the dialog, check the option Regular expressions and close the dialog again. Select the text you want to search for and reopen the dialog: the Find field will now contain the regular expression for the text to find.

- **Find**: Click to find the next instance of the source string.
- **Replace/Find**: Click to replace the current instance with the replacement text and go to the next instance of the source string.
- **Replace**: Click to replace the current instance with the replacement text.
- **Replace All**: Click to replace all instances of the source string with the replacement text.
- **Close**: Close the dialog.

Font Manager

The Fonts Manager contains the fonts that were added to the template manually. It essentially lists the fonts located in the **Fonts** folder of the Resources pane (see "Fonts" on page 332).

Fonts with the same file name with a different extension are considered variations of the same font. For example, if there are three files, named gotham-book-webfont.eot, gotham-book-webfont.ttf, gotham-book-webfont.woff, only "gotham-book-webfont" appears in the Name column of this dialog.

The following buttons appear to the right of the list of fonts:

- **New**: Click to open the Edit Font dialog to add a new font.
- **Edit**: Click to open the Edit Font dialog to edit the currently selected font.
- **Remove**: Click to delete the currently selected font entry.
- **Duplicate**: Click to create a copy of the currently selected font entry.

Edit Font

The Edit Font dialog appears when clicking New or Edit from the Fonts Dialog.

- **Name**: Enter the name that should be used to refer to the font. This is equivalent to the `font-family` property of the @font-face CSS rule (see https://www.w3schools.com/cssref/css3_pr_font-face_rule.asp).
- **Font Weight**: Use the drop-down to select the default font weight (the thickness, see https://www.w3schools.com/cssref/pr_font_weight.asp):
  - **None**: Does not define the property.
  - **Normal**: Defines font-weight as normal
  - **Bold**: Defines the font-weight as bold (equivalent to a numerical value of 700).
  - **Numerical values**: Defines the line thickness; 400 is normal, 700 is bold.
- **Font Style**: Use the drop-down to select the font style (see https://www.w3schools.com/cssref/pr_font_font-style.asp:
  - **None**: Does not define the property.
  - **Normal**: Defines font-style as normal
  - **Italic**: Makes the font italic.
  - **Oblique**: Makes the font oblique (this is generally the same as italic but does not require a special italic version of the font).
- **Name**: Check the fonts in the list to include them in the font definition.

Image Formatting dialog

The Image Formatting dialog is accessible by selecting an image in the template and then selecting **Format > Image** in the menu.

All settings in this dialog are in fact CSS properties. Cascading Style Sheets (CSS) were originally designed for use with web pages: HTML files. Since Designer templates are HTML files, they are styled with CSS. To learn how to use CSS in the Designer, see "Styling and

formatting" on page 294 and "Styling templates with CSS files" on page 296. For information about specific properties and their options, see W3Schools CSS Reference.

For more information about the use of images, see "Images" on page 271 and "Styling an image" on page 320.

Image Tab

- **General group**:
  - **Width**: Set the width of the image in measure or percentage. Equivalent to the CSS `width` property.
  - **Height**: Set the height of the image in measure or percentage. Equivalent to the CSS `height` property.
  - **Angle**: Set the rotation angle of the image in clockwise degrees. Equivalent to the CSS `transform:rotate` property.
  - **Corner radius**: Set the radius of rounded border corners in measure or percentage. Equivalent to the CSS `border-radius` property.
  - **Display**: Use the drop-down or type in the value for how to display the image. Equivalent to the CSS `display` property.
  - **Overflow**: Use the drop-down or type in the value for how to handle overflow (the part of the image that does not fit in the current size of the box). Equivalent to the CSS `overflow` property.
  - **Source**: Enter the web address or local file address of the image. Equivalent to the HTML `src` attribute.
  - **Alternate text**: Enter an alternate text for the image. This is displayed in browsers and email clients when the image is loading or if the image cannot be displayed. It is also used for accessibility. Equivalent to the HTML `alt` attribute.
- **Text wrap group**:
  - **Float**: Use the drop-down or type in the value for how to float the image, if the image is not in an absolute position (see Position, below). Equivalent to the CSS `float` property.
  - **Clear**: Use the drop-down or type the value to clear pre-existing alignments. Equivalent to the CSS `clear` property.
- **Positioning**:
  - **Position**: Use the drop-down or type in the value for the type of positioning for the image. Equivalent to the CSS `position` property (see "Using the CSS position

property" on page 314).

- **Top**: Set the vertical offset between this image and its parent's top position. Equivalent to the CSS `top` property.
- **Left**: Set the horizontal offset between this image and its parent's left position. Equivalent to the CSS `left` property.
- **Bottom**: Set the vertical offset between this image and its parent's bottom position. Equivalent to the CSS `bottom` property.
- **Right**: Set the horizontal offset between this image and its parent's right position. Equivalent to the CSS `right` property.
- **Z-index**: Set the z-index of the image. The z-index defines in which order elements appear. Equivalent to the CSS `z-index` property.

Spacing Tab

- **Padding group**: Defines padding (spacing inside the element) in measure or percentage:
  - **All sides**: Check to set all padding to use the Top value. Equivalent to the CSS `border` property.
  - **Top, Left, Bottom, Right**: Set padding for each side. Equivalent to the CSS `border-left`, `border-top`, `border-right` and `border-bottom` properties.
- **Margin group**: Defines margins (spacing outside the element) in measure or percentage:
  - **All sides**: Check to set all margins to use the Top value. Equivalent to the CSS`margin`property.
  - **Top, Left, Bottom, Right**: Set the margin for each side. Equivalent to the CSS `margin-left`, `margin-top`, `margin-right` and `margin-bottom` properties.

Border Tab

- **Same for all sides**: Defines the border properties for all sides using the Top properties. Equivalent to the CSS `border` property.
- **Top, Left, Bottom, Right**: Each group defines the following properties:
  - **Width**: Specify the thickness of the border. Equivalent to the CSS `border-width` property.
  - **Style**: Specify the style of the border such as `solid`, `dashed` or `dotted`. Equivalent to the CSS `border-style`property.

- **Color**: Specify the color of the border: select a named color (defined in the "Colors Properties" on page 492) from the drop-down, or click the colored square to open the Color Picker dialog ("Color Picker" on page 492). Alternatively you could type a name or value in the Color field directly. It must be a predefined CSS color name (CSS color names), a hexadecimal color code (HTML Hex Color), an RGB color value, for example `rgb(216,255,170)` or a CMYK color value, for example `cmyk(15%, 0%, 33%, 0%)`. Equivalent to the CSS `border-color` property.

Import Resources dialog

The Import Resources dialog lets you import a variety of resources from other templates.

To open this dialog, click **File > Import Resources...** in the menu.

1. Select the template to copy resources from and click **OK**.
2. Now you can select various resources: "Master Pages" on page 156, "Media" on page 159, "Sections" on page 119, "Images" on page 271, style sheets (see "Styling templates with CSS files" on page 296), as well as scripts (see "Scripts pane" on page 590) and the Data Model (see "Data Model pane" on page 579).
   If you want to select individual items instead of all resources in a certain folder, expand the folder using the arrow to the left.

   > **Note**
   >
   > Resources may be interdependent. A Print section may need a certain Master Page or Media, for example; a Dynamic Image script relies on the presence of certain images; etcetera. This isn't visible in the Import Resources dialog. Make sure to select all necessary resources.

3. Choose what happens when a resource with the same name at the same hierarchy level already exists in your template.
   Names that represent files or folders on disk are case-insensitive, while all other names are case-sensitive.
   - **Rename**: Import the resource, giving its name a suffix to make it unique in the target folder.
     Note that when a section is renamed, the properties of its context are not copied over, except for Includes (Web contexts only; see Includes). Includes are transferred

to the imported section.

- **Overwrite**: Replace the existing resource.
  If the imported resource is a section, the properties of its context are copied to the target Context. For a Print context this can include settings for Color Output and Finishing.
  When replacing a resource that is opened in a separate editor (not the main editor), that editor will be closed and any unsaved changes will be lost.
- **Skip**: Don't import the resource.

> **Note**
>
> When it's selected for import, the Data Model is always overwritten, regardless of the policy for duplicates.

4. Click OK to import the resources.
5. Verify that the template works as expected. Make sure that the template contains everything that the imported resources need, and that all context and section settings are correct. Doing a Preflight is recommended (see "Doing a Preflight" on page 405).

Includes dialog

The Includes dialog defines which style sheets and JavaScript files should be applied to a section when generating output (see: "Styling templates with CSS files" on page 296 and Using JavaScript).

To open this dialog and make settings for one section, right-click the section on the **Resources** pane and select **Includes**.

Email clients do not read CSS files and some even remove a <style> tag when it is present in the email's header. Nevertheless, CSS files can be used with the Email context in the Designer.
When generating output from the Email context, all CSS rules that apply to the content of the email can be processed and added either to the header of the email or to inline style properties as if local formatting was applied, depending on the Email section properties. See "Email section properties" on page 540.

1. From the **File types** dropdown, select **Stylesheets**, **JavaScripts** or **all**.
2. The list at the left displays the style sheets and/or JavaScript files that are present in the template's resources. The list at the right shows the style sheets and or JavaScript files that will be included in the output of the current section (or in all Web sections, if you are making settings for the Web context). Use the **Include** and **Exclude** buttons to move files from one list to the other.
3. Files are included in the order shown. To change this order, click one of the included files and use the **Up** or **Down** button.

> **Note**
>
> The styles in each following style sheet add up to the styles found in previously read style sheets. When style sheets have a conflicting rule for the same element, class or ID, the **last** style sheet 'wins' and overrides the rule found in the previous style sheet.

JavaScript Properties

The JavaScript Properties dialog appears when you right-click a JavaScript file on the **Resources** pane and select **Properties**. (See also: Using JavaScript.)

It contains three options:

- **Name**: This option lets you rename the JavaScript resource in the template. The link between the JavaScript file and Web sections that include it will not be broken by renaming the file. (See also: Including a JavaScript file in a Web context.)
- **Async**: When **async** is checked, the script executes asynchronously with the rest of the page (while the page continues the parsing).
- **Defer**: This option postpones the execution of the script until the page has finished parsing. This is required by APIs like Google Maps.

It is recommended to check at least one of the options: async or defer, or both. Page content will become visible more quickly, resulting in a better user experience.
If you enable **both** options, async takes precedence on modern browsers, while older browsers that support defer but not async will fallback to defer.
When **neither** of the options is checked, the script gets fetched and executed immediately, interrupting the page loading and rendering.

JSON sample data dialog

The JSON sample data dialog lets you load JSON data into the Data Model (see "Loading data" on page 341).
By default, the JSON data is mapped to corresponding fields in the existing Data Model. Data that does not correspond to any field is discarded.
You may use the Replace Data Model option to let the JSON data replace the existing Data Model.

The dialog is opened via the menu: **File > Add data > JSON sample data**, or via the JSON Sample Data toolbar button on the Data Model pane.

The options in this dialog are:

- **File**: The path and name of the JSON file to use. The file is assumed to be UTF-8 encoded.
- **Browse**: Opens an explorer window to browse folders and select a JSON file.
- The box below the file name allows to paste or enter JSON data. After opening a JSON file the JSON data will appear in this box. You can review and edit the JSON.
- **Replace Data Model**: Selecting the Replace Data Model option removes the existing Data Model from the template and creates a new Data Model based on the keys found in the JSON. When this option is not checked, the JSON data is mapped to any corresponding fields in the existing Data Model, and data that does not correspond to any field is discarded.
- **Finish**: If the JSON is valid, you may click Finish to import the data into the Data Model pane. Arrays of objects are converted to records; key-value pairs are converted to data fields. If a value in a record object is a JSON object, it is ignored.

> **Note**
>
> The JSON Sample Data is not stored with the template. In order to reuse the data you should create a JSON file on disk and load it via the **File** option in this dialog.

Types of JSON data

Examples

A single record with two fields

```
{
"first": "Peter",
"last": "Parker"
}
```

Two records

```
[{
"first": "Peter",
"last": "Parker"
},
{"first": "Martin",
"last": "Moore"
}]
```

A single record with a detail table

```
{
"name":"Peter Parker",
"email":"parkerp@localhostcom",
"detail": [{"id":"inv123","ExtraData":"hello"},
{"id":"456","ExtraData":"world"}]
}
```

A JSON Record Data List

A JSON Record Data List describes a list of data fields (as name/value pairs), a data table schema and nested data records (if any) for one or more data records. This example holds a single, simple record, and two runtime parameters.

```
{
"parameters": {
        "campaign":"Campaign Name",
        "folder":"myFolder"
        },
"data": {
        "schema": {
                "columns": {
                "ID": "STRING",
                "Gender": "STRING",
```

```
                "FirstName": "STRING",
                "LastName": "STRING"
                }
        },
        "fields": {
                "ID": "CU00048376",
                "Gender": "M.",
                "FirstName": "Benjamin",
                "LastName": "Verret"
                }
        }
}
```

Note that the "parameters" key can be left out if there are no runtime parameters.

For more examples see the REST API Cookbook, and Working with JSON data in the Workflow Online Help.

Line Chart Properties dialog

The Line Chart dialog appears when a Line Chart is right-clicked and the **Chart...** option is clicked. It determines how the chart is displayed when generating output and in Preview mode (see "Business graphics" on page 260).

General tab

Most settings on the General tab correspond to properties of the AmSerialChart class in the amCharts library; see: AmSerialChart.

- **General Group**:

  Display grid above graph: Check to display the grid on top of the bars so that it is always visible. (Equivalent to the
  - `gridAboveGraphs`
  property; see: gridAboveGraphs.)

  Rotate: Check to rotate the chart 90 degrees so that the bars are horizontal starting from the left. (Equivalent to the
  - `rotate`
  property; see: rotate.)

  Stack Series: Check to stack the values in one bar, instead of having one bar per value. (Equivalent to the
  - `stackType`

property of the
`ValueAxis`
; see: stackType.)

- **Text Group**: Determines how text is displayed in labels and legends.

  Font: Enter the font-face to use to display text. The font must be installed on the system and defaults to Verdana if the font is not found. (Equivalent to the
  - `fontFamily`
  property; see: fontFamily.)

  Size: Enter the size of the font. Defaults to 11. (Equivalent to the
  - `fontSize`
  property; see: fontSize.)

  Color: Select the color in which to display text: click the colored square to open the Color Picker dialog ("Color Picker" on page 492), or enter a valid hexademical color (HTML Hex Color) or a predefined CSS color (CSS color names). (Equivalent to the
  - `color`
  property; see: color.)

- **3D Group**: Creates a 3D effect if both settings in this group are higher than 0.

  Depth: The depth of the 3D part of the plot area. (Equivalent to the
  - `depth3D`
  property; see: depth3D.)

  Angle: The angle of the 3D part of the plot area. (Equivalent to the
  - `angle`
  property; see: angle.)

Value Axis tab

Settings on the Value Axis tab are equivalent to properties of the ValueAxis class in the amCharts library; see: ValueAxis.

- **Title group**:

  Label: Enter a label for the value axis (the Y axis, or the X axis if the graph is rotated). (Equivalent to the
  - `title`
  property; see: title.)

  Bold: Check this option if the title should be bold. (Equivalent to the
  - `titleBold`
  property; see: titleBold.)

Color: The default label color is black. To select a custom color for the label, click the colored square to open the Color Picker dialog ("Color Picker" on page 492), or enter a valid hexademical color (HTML Hex Color) or a predefined CSS color (CSS color names). (Equivalent to the

- `titleColor`
  property; see: titleColor.)

Font Size: Enter a font size for the label. (Equivalent to the

- `titleFontSize`
  property; see: titleFontSize.)

- **Grid group**:

  Color: Select a color for the grid lines that divide the value axis: click the colored square to open the Color Picker dialog ("Color Picker" on page 492), or enter a valid hexademical color (HTML Hex Color) or a predefined CSS color (CSS color names). (Equivalent to the

  - `gridColor`
    property; see: gridColor.)

  Opacity: Enter the opacity percentage of the grid. Default is 15%. 100 is fully opaque, 0 is transparent. (Equivalent to the

  - `gridAlpha`
    property; see: gridAlpha.)

  Thickness: Enter a thickness for the grid lines. Default is 1. (Equivalent to the

  - `gridThickness`
    property; see: gridThickness.)

  Tick Length: Length of the tick marks. (Equivalent to the

  - `tickLength`
    property; see: tickLength.)

- **Axis group**:

  Color: Select a color for the value axis: click the colored square to open the Color Picker dialog ("Color Picker" on page 492), or enter a valid hexademical color (HTML Hex Color) or a predefined CSS color (CSS color names). (Equivalent to the

  - `axisColor`
    property; see: axisColor.)

  Opacity: Enter the opacity in percentage for the axis. 100 is fully opaque, 0 is transparent. Set opacity to 0 to hide the axis. (Equivalent to the

  - `axisAlpha`
    property of the
    `ValueAxis`

; see: axisAlpha.)

Thickness: Enter the thickness of the axis. (Equivalent to the

- `axisThickness`
  property of the
  `ValueAxis`
  ; see: axisThickness.)

Labels: Check this option to make the labels on the value axis visible. This is often useful in rotated charts. (Equivalent to the

- `labelsEnabled`
  property of the
  `ValueAxis`
  ; see: labelsEnabled.)

Frequency: Defines per how many grid lines a label will be placed. (Equivalent to the

- `labelFrequency`
  property of the
  `ValueAxis`
  ; see: labelFrequency.)

Category Axis tab

Settings on the Category Axis tab correspond to properties of the CategoryAxis class in the amCharts library; see: CategoryAxis.

- **Title group**:

  Label: Enter a label for the category axis (the X axis, or the Y axis if the graph is rotated).(Equivalent to the

  - `title`
    property; see: title.)

  Bold: Check if you want the title to be bold. (Equivalent to the

  - `titleBold`
    property; see: titleBold.)

  Color: Select a custom color for the label (default is black): click the colored square to open the Color Picker dialog ("Color Picker" on page 492), or enter a valid hexademical color (HTML Hex Color) or a predefined CSS color (CSS color names). (Equivalent to the

  - `titleColor`
    property; see: titleColor.)

  Font Size: Enter a font size for the title. (Equivalent to the

  - `titleFontSize`

property; see: titleFontSize.)

- **Grid group**:

  Color: Select a color for the grid lines that divide the category axis: click the colored square to open the Color Picker dialog ("Color Picker" on page 492), or enter a valid hexademical color (HTML Hex Color) or a predefined CSS color (CSS color names). (Equivalent to the
  - `gridColor`
  property; see: gridColor.)

  Opacity: Enter the opacity percentage of the grid. Default is 15%. 100 is fully opaque, 0 is transparent. (Equivalent to the
  - `gridAlpha`
  property; see: gridAlpha.)

  Thickness: Enter a thickness for the grid lines. Default is 1. (Equivalent to the
  - `gridThickness`
  property; see: gridThickness.)

  Position: Specifies if a grid line is placed on the centre of a cell or on the beginning of a cell. (Equivalent to the
  - `gridPosition`
  property; see: gridPosition.)

  > **Tip**
  >
  > The position of the ticks (which is Middle by default) does not move with the grid. To change the tick position, add the following to the `categoryAxis`
  >
  > section on the Source tab:
  > `"tickPosition": "start"`
  >
  > .

  Tick Length: Length of the tick marks. (Equivalent to the
  - `tickLength`
  property; see: tickLength.)

> **Tip**
>
> By default, one label per grid line will appear on the Category axis. To change that frequency, add the following to the `categoryAxis`
>
> section on the Source tab: `"labelFrequency": "2"`
>
> (replace 2 by the desired number of grid lines).

- **Axis group**:

  Color: Select a color for the value axis: click the colored square to open the Color Picker dialog ("Color Picker" on page 492), or enter a valid hexademical color (HTML Hex Color) or a predefined CSS color (CSS color names). (Equivalent to the
  - `axisColor` property; see: axisColor.)

  Opacity: Enter the opacity in percentage for the axis. 100 is fully opaque, 0 is transparent. Set opacity to 0 to hide the axis. (Equivalent to the
  - `axisAlpha` property; see: axisAlpha.)

  Thickness: Enter the thickness of the axis. (Equivalent to the
  - `axisThickness` property; see: axisThickness.)

  Auto Wrap: Specifies if axis labels (when horizontal) should be wrapped if they don't fit in the allocated space. (Equivalent to the
  - `autoWrap` property; see: autoWrap.)

Graphs

Settings on the Graphs tab correspond to properties of the `AmGraph` class in the amCharts library; see: AmGraph.

- **Line**:
  - **Thickness**: Sets the thickness of the lines in a Line chart. (Equivalent to the `lineThickness` property; see: lineThickness.)

Tip

Set `"fillAlphas" = 1` on the Source tab to color the space below the lines.

Legend tab

Settings on the Legend tab correspond to properties of the AmLegend class in the amCharts library; see: AmLegend.

- **Show Legend**: Specifies if legend is enabled or not. (Equivalent to the `enabled` property; see: enabled.)
- **Legend Group**: Defines how the legends are shown.

  Equal label widths: Check so that all labels are of equal width in the Legends box. The Legend's width will accommodate the largest value. (Equivalent to the `equalWidths` property; see: equalWidths.)

  Position: Use the drop-down to select where the legend is shown: at the Right, Left, Top or Bottom. (Equivalent to the `position` property; see: position.)

  Align: Use the drop-down to select how to align the text in the labels: Left, Middle or Right. (Equivalent to the `align` property; see: align.)

  Horizontal Space: Horizontal space between legend items, in pixels. (Equivalent to the `spacing` property; see: spacing.)

  Vertical Space: Enter a numerical value (in pixels) to define the vertical space between legend items, and also between the legend border and the first and last legend item. (Equivalent to the `verticalGap` property; see: verticalGap.)

  Max Columns: Enter a numerical value to define the maximum number of columns in the legend. If the Legend's position is set to "right" or "left", this is automatically set to 1. (Equivalent to the `maxColumns`

property; see: maxColumns.)

- **Labels Group**: Defines if and how labels are shown in the Legend.

  Text: Enter the text used to display the labels;
  - `[[title]]`
    is a variable that will be replaced with the title of the graph. (Equivalent to the `labelText`
    property; see: labelText.)

- **Markers Group**: Defines how the Legend's Markers look. Markers are icons with a color matching the Legend with its corresponding line.

  Type: Use the drop-down to select in which shape the Markers are displayed; "none" hides the Markers completely. (Equivalent to the
  - `markerType`
    property; see: markerType.)

  Size: Enter the size (in pixels) for the Markers to be displayed. (Equivalent to the
  - `markerSize`
    property; see: markerSize.)

  Label Gap: Enter the distance (in pixels) between the legend marker and legend text. (Equivalent to the
  - `markerLabelGap`
    property; see: markerLabelGap.)

  Border Width: Use the drop-down to define the thickness of the border added to the Markers. The default value (0) means the line will be a "hairline" (1 px). In case the Marker type is line, this style will be used for the line thickness. (Equivalent to the
  - `markerBorderThickness`
    property; see: markerBorderThickness.)

  Border Color: Color of the Legend's border. Enter a valid hexademical color (HTML Hex Color) or a predefined CSS color (CSS color names), or click the colored square to open the Color Picker dialog ("Color Picker" on page 492). (Equivalent to the
  - `markerBorderColor`
    property; see: markerBorderColor.)

  Border Opacity: Enter a numerical value between 0 and 100 to define the opacity (in percentage) of the border. (Equivalent to the
  - `markerBorderAlpha`
    property; see: markerBorderAlpha. When specified on the Source tab, the value should be between 0 and 1, e.g. 0.8.)

Source tab

The JSON on the Source tab reflects the choices made in the other tabs and, more importantly, provides the possibility to add in any amCharts configuration option that is unavailable via the other tab menus. For more information see: "Adding and editing properties manually" on page 265.

Locale Settings

The Locale dialog box sets the locale used inside the template. The Locale can affect time, currency output, and other formatting that depends on location and language (see "Locale" on page 336). The default Locale for new templates can be set via the Preferences ("Language preferences" on page 386).

- **Use**: Use the drop-down to select how the Locale is set for the current template.
  - **System Locale**: Select this to use the operating system's locale settings. This is set in the Region settings of the control panel.
  - **Explicit Locale**: Select this option to specify a static locale which will remain static for this template, whichever server the template is used on.
  - **Data Field**: Select this to use a data field from the record. The locale will be record-specific in this case.
  - **Runtime Parameter**: Select this to use the value of a runtime parameter. Runtime parameters are defined in the template, but their value is set at runtime; see Runtime parameters.
- **Locale**: Use the drop-down to select a specific locale. Only enabled when **Explicit Locale** is selected above.
- **Data Field**: Use the drop-down to select a field within the current data model that contains the locale.
- **Runtime Parameter**: Use the drop-down to select a runtime parameter that will contain the locale.

Master Page Properties

Master Pages can only be used in a Print context; see "Master Pages" on page 156.

The following properties are available for Master Page resources:

- **Name**: The name of the master page, displayed in all drop-downs where the Master Page is shown as well as in the "Resources pane" on page 583.
- **Margins** group:
  - **Header**: The space at the top of the Master Page where no content will print, when this Master Page is used in a Section.
  - **Footer**: The space at the bottom of the Master Page where no content will print, when this Master Page is used in a Section.

Media Properties

Media can only be used in a Print context. For an explanation of what Media are and how to use them, see "Media" on page 159.

Media are not printed, unless you want them to; see "Printing virtual stationery" on page 166.

To open the Media properties dialog, right-click one of the Media in the Media folder on the **Resources** pane and select **Properties**.

The **Page Setup** button closes the Media Properties dialog and opens the Print Section Properties dialog (see "Print section properties" on page 542).

Properties Tab

- **Name**: The name of the Media, displayed in all drop-downs where the Media is shown as well as in the Resources Pane.
- **Size**: This group is read-only and only used to display the size selected in the linked Print section's properties (see "Print section properties" on page 542).
  - **Page Size**: The named page size.
  - **Width**: The width of the page.
  - **Height**: The height of the page.

Virtual Stationery Tab

- **Front/Back group**: Defines the pre-printed media used for the front and back of the Virtual Stationery.
  - **PDF**: Click the **Select Image** button to open the "Select Image dialog" on page 545 and select which PDF (and optionally, which page of the PDF) to display as a

background for the page.
The **Clear** button can be used to clear the path. If the path is not empty it will be used when the Media is linked to a section via a Control script.

- **Position**: Use the drop-down to select how the PDF is displayed on the page:
  - **Fit to Media**: Select to stretch the PDF to fit the media size.
  - **Centered**: Select to center the PDF on the page, vertically and horizontally.
  - **Absolute**: Select to place the PDF at a specific location on the page. Use the Top and Left options below to specify the positioning of the PDF.
    - **Top**: The distance between the top side of the page and the top side of the PDF.
    - **Left**: The distance between the left side of the page and the left side of the PDF.

- **Front side**: Select the image that is shown as a background for all "front" sides in the template.
- **Back side**: Select the image that is shown as a background for all "back" sides in the template.

Characteristics tab

The characteristics define the type of paper on which the Print context is meant to be printed.

- **Media Type**: The type of paper, such as *Continuous*, *Envelope*, *Labels*, *Stationery*, etc.
- **Weight**: The weight of the media in grammage (g/m2).
- **Front Coating**: The pre-process coating applied to the front surface of the media, such as *Glossy*, *High Gloss*, *Matte*, *Satin*, etc.
- **Back Coating**: The pre-process coating applied to the front surface of the media.
- **Texture**: The texture of the media, such as *Antique*, *Linen*, *Stipple* or *Vellum*.
- **Grade**: The grade of the media, such as *Gloss-coated paper*, *Uncoated white paper*, etc.
- **Hole Name**: Pre-defined hole pattern that specifies the pre-punched holes in the media, such as *R2-generic*, *R2m-MIB*, *R4i-US*, etc.

New dialog

The New dialog lets you create a new OL Connect project, data mapping configuration or template.

To open this dialog, select **File > New** from the menu or click the **New** toolbar button.

Options for this dialog:

- **Wizards**: Type a text to filter the templates. (See "Templates" on page 109.)
- **Data Mapping Configurations**: Lets you create a new data mapping configuration for a database or data file.
  - **Databases**: Select the database type and click **Next**. For the options on the next page, see Using the wizard for databases.
  - **Files**: Select the file type and click **Next**. Then select the file itself and click **Finish**. After opening the file, you have to make settings for the input data (see Data Source settings) to make sure that the source data is parsed correctly and divided into logical units of data the way you want. Then you can start building the data extraction workflow.
- **Data mapping Wizards**: Lets you create a new data mapping configuration with a wizard. See With a wizard.
- **Sample Projects**: Lists all Sample Projects. A Sample Project generates a Connect solution consisting of a Workflow configuration as well as any templates, data mapping configurations, and Print Presets that are used in that configuration.
- **Templates**: Lists all types of templates that you can create with a wizard. See "Templates" on page 109.
  - **Email Template**: This starts the Basic Action Email wizard; see "Creating an Email template with a Wizard" on page 171.
  - **PDF-based Print Template**: The PDF-based Print template wizard creates a document from an existing PDF file; see "PDF-based Print template" on page 129.
  - **Print Template**: This starts the Formal Letter Print wizard; see "Creating a Print template with a Wizard" on page 126.
  - **Web Page Template**: This starts the most basic Web Page wizard; see Creating a Web template with a Wizard.
  - **Word-based Print Template**: This starts the Word-based Print wizard; see "Creating a Print template with a Wizard" on page 126.
  - **Banded Email Templates**: The contents of the email templates that these wizards create extend to the edge of the window in which it is read. See "Basic Email and Banded Email" on page 174.
  - **Basic Email Templates**: The contents of the email templates that these wizards create extend to the email's margin. See "Basic Email and Banded Email" on page 174.

- **Basic Print Templates**: From this folder you can run the Formal Letter or Postcard Print wizard; see "Creating a Print template with a Wizard" on page 126.
- **Capture OnTheGo Starter Templates**: COTG templates are used to generate forms for the Capture OnTheGo mobile application. See Capture OnTheGo template wizards.
- **ERP Templates**: ERP template wizards are used to create business documents, such as a sales invoice, purchase order, collection letter, etc.. See "ERP templates" on page 131.
- **Foundation Web Page Starter Templates**: These template wizards create Web templates that make use of the Zurb **Foundation** front-end framework. See Creating a Web template with a Wizard.
- **Slate: Responsive Email Templates by Litmus**: These template wizards create responsive HTML email templates. See "Slate: Responsive Email Templates by Litmus" on page 173.

> **Note**
>
> When you create a new project you are prompted to specify a folder for the sample project. You may also be required to enter your email address to which will be used to identify your changes in the project history.

Package dialog

The Package dialog saves templates and Print Presets as a package file. Package files can be sent to other Connect Designer users.

- **Configurations**: Initially this list contains the Connect files that are currently open.
  - **Browse**: This button opens the Select Files dialog that lets you add one or more templates (*.OL-template) and Print Presets (*.OL-jobpreset, *.OL-outputpreset) to the list.
  - **Select Preset**: Opens the Select Preset dialog. This dialog lists all Print Presets found in the folder: `<root>\Users\<username>\Connect\workspace\configurations`. There are two ways to make it easier to find the file that you need:
    - The **Show** drop-down list lets you select the type of preset that you want to open.

- In the **Filter** field, you can enter (part of) a preset name to filter the list for matching presets.

  Select one or more Print Presets to add them to the list of configurations in the Send to Workflow dialog.
- **Delete**: Deletes the selected file from the list of configurations.

- **Package...**: This button opens the Save dialog. All selected Connect files will be saved to one package file. You can use the topmost checkbox to select all of the listed files at once.
- **Cancel**: Click this button to close the dialog without creating a package file.

Paragraph Formatting dialog

The Paragraph Formatting dialog controls how the selected paragraph is formatted. It is accessed by placing the cursor within a paragraph and then selecting **Format > Paragraph** on the menu.

All settings in this dialog are in fact CSS properties. Cascading Style Sheets (CSS) were originally designed for use with web pages: HTML files. Since Designer templates are HTML files, they are styled with CSS. To learn how to use CSS in the Designer, see "Styling and formatting" on page 294 and "Styling templates with CSS files" on page 296. For information about specific properties and their options, see W3Schools CSS Reference.

Click the **Advanced** button to enter CSS properties and values directly.

The **Apply** button lets you preview the template with the new settings.

For information about text and how to style it see "Text and special characters" on page 283 and "Styling text and paragraphs" on page 308.

Formats tab

- **General group**:
  - **Line-height**: Specify the height of each line in the element's text, in measure or percentage. Note that this is not spacing between lines, but rather the complete height of the line itself including the text. Equivalent to the `line-height` property in CSS.
  - **Align**: Select how text should be aligned, such as `left`, `center`, `right` or `justify`. Equivalent to the `align` property in CSS.

- **First Indent**: Specify the indentation of the first line of each paragraph in the element. Equivalent to the `text-indent` property in CSS.
- **Display**: Select how to display the element. This can also be used to hide an element completely using the `none` option. See CSS Display. Equivalent to the `display` property.
- **Direction**: Select in which direction text should be displayed (ltr, rtl, auto). Useful for certain languages such as arabic, hebrew, etc. Equivalent to the `dir` HTML attribute.

- **Breaks group**:
  - **Before**: Specifies whether a page break should occur **before** the paragraph. This is equivalent to the `page-break-before` property in CSS; see CSS page-break-before property for an explanation of the available options.
  - **Inside**: Specifies whether a page break is allowed inside the paragraph. Equivalent to the `page-break-inside` property in CSS; see CSS page-break-inside property for an explanation of the available options.
  - **After**: Specifies whether a page break should occur **after** the paragraph. Equivalent to the page-break-after property in CSS; see CSS page-break-after property for an explanation of the available options.
  - **Widows**: Specifies how to handle widows within the paragraph (lines appearing alone on the next page if the paragraph does not fit on the current one). Equivalent to the `widows` property. Widows and orphans are ignored if the `page-break-inside` property is set to `avoid`.
  - **Orphans**: Specifies how to handle orphans within the paragraph (lines appearing alone at the end of a page if the paragraph does not fit on the current one). Equivalent to the `orphans` property.

> **Note**
>
> For more information on page breaks, widows and orphans, see the W3 Paged Media reference.

Spacing tab

- **Padding group**: Defines padding (spacing inside the element) in measure or percentage:

- **All sides**: Check to set all padding to use the Top value. Equivalent to the CSS `padding` property.
- **Top, Left, Bottom, Right**: Set padding for each side. Equivalent to the CSS `padding-left`, `padding-top`, `padding-right` and `padding-bottom` properties.

- **Margin group**: Defines margins (spacing outside the element) in measure or percentage:
  - **All sides**: Check to set all margins to use the Top value. Equivalent to the `margin` property.
  - **Top, Left, Bottom, Right**: Set the margin for each side. Equivalent to the `margin-left`, `margin-top`, `margin-right` and `margin-bottom` properties.

Border tab

- **Same for all sides**: Defines the border properties for all sides using the Top properties. Equivalent to the `border` property.
- **Top, Left, Bottom, Right**: Each group defines the following properties:
  - **Width**: Specify the thickness of the border. Equivalent to the `border-width` property.
  - **Style**: Specify the style of the border such as `solid`, `dashed` or `dotted`. Equivalent to the `border-style` property.
  - **Color**: Specify the color of the border. select a named color (defined in the "Colors Properties" on page 492) from the drop-down, or click the colored square to open the Color Picker dialog ("Color Picker" on page 492). Alternatively you could type a name or value in the Color field directly. It must be a predefined CSS color name (CSS color names), a hexadecimal color code (HTML Hex Color), an RGB color value, for example `rgb(216,255,170)` or a CMYK color value, for example `cmyk(15%, 0%, 33%, 0%)`. Equivalent to the `border-color` property.

PDF Attachments dialog

The PDF Attachments dialog defines options for the Email context that are used when generating email output with PDF attachments (see "Generating Email output" on page 1017).

To open this dialog, right-click the **Email** context on the **Resources** pane and select **PDF attachments**.
Alternatively, select **Context > PDF Attachments** on the main menu. This option is only available when an Email section is being edited in the Workspace.

- **Print Context Image Compression**: Defines the properties of the PDF when attaching the Print context to email output.
  - **Lossless**: Enables maximum quality in the PDF. Note that this will produce a larger PDF.
  - **Quality**: Disabled when Lossless is checked. Determines the quality (aka compression) of the attached PDF.
  - **Tile Size**: Use the drop-down to select the size of the tiles used in the image. When low Quality values are used to optimize images smaller than 1024 x 1024 pixels, using the largest tile size will produce better results.

Pie Chart Properties dialog

The Pie Chart Properties dialog appears when a Pie Chart is right-clicked and the **Chart...** option is clicked. It determines how the Pie Chart is displayed when generating output and in Preview mode (see "Business graphics" on page 260).

General tab

Settings on the General tab correspond to properties of the `AmPieChart` class in the amCharts library; see: AmPieChart.

- **Text Group**: Determines how text is displayed in labels and legends.
  - **Font**: Type in the font-face to use to display text. The font must be installed on the system and defaults to Verdana if the font is not found. (Equivalent to the `fontFamily` property; see: fontFamily.)
  - **Size**: Type in the size of the font. Defaults to 11. (Equivalent to the `fontSize` property; see: fontSize.)
  - **Color**: Type the color in which to display text: a valid hexademical color (HTML Hex Color) or a predefined CSS color (CSS color names), or click the colored square to open the Color Picker dialog ("Color Picker" on page 492). (Equivalent to the `color` property; see: color.)
- **Slice Colors Group**

Note

Settings made in this group override the settings made in the Chart Wizard (see "Selecting data for a Business Graphic" on page 263).

- **Base Color**: Enter a base color: a valid HTML hexadecimal color value (HTML Hex Color) or a predefined CSS color (CSS color names), or click the colored square to open the Color Picker dialog ("Color Picker" on page 492). When a Base Color is set, it will be the color of the first slice and the colors of all other slices are based on this color. (Equivalent to the `baseColor` property; see: baseColor.)
  To set a color for each data field individually, leave this option empty and edit the **script** that fills the Pie Chart (see "Selecting data for a Business Graphic" on page 263).
- **Brightness Step**: Enter the amount of brightness to change on each new slice. Positive values increase the brightness (max: 100), negative values decrease the brightness (minimum: -100). Default is 10. (Equivalent to the `brightnessStep` property; see: brightnessStep.)

- **Slice Outline Group**: Determines whether an outline should be added to each slice of the chart.
  - **Width**: Select the width of the outline for each pie slice. (Equivalent to the `outlineThickness` property; see: outlineThickness.)
  - **Color**: Enter a color for the outline: a valid HTML hexadecimal color value (HTML Hex Color) or a predefined CSS color (CSS color names), or click the colored square to open the Color Picker dialog ("Color Picker" on page 492). (Equivalent to the `outlineColor` property; see: outlineColor.)
  - **Opacity**: Enter the opacity of the outline. 100 is fully opaque, 0 is transparent. (Equivalent to the `outlineAlpha` property; see: outlineAlpha.)

Pie tab

Settings on the Pie tab correspond to properties of the `AmPieChart` class in the amCharts library; see: AmPieChart.

- **Pie Group**: Defines how the pie chart is displayed in the template.
  - **Radius**: Enter the radius of the Pie Chart as a percentage of the shortest length of the containing object (e.g. 30%) or in pixels. (Equivalent to the `radius` property; see: radius.)

- **Hole Radius**: Enter the radius of the center of the Pie Chart to remove, in pixels or as a percentage. The hole radius removes the center of the chart, creating a doughnut hole pie chart. (Equivalent to the `innerRadius` property; see: innerRadius.)
- **Start Angle**: Enter the starting angle of the first slice of the chart, between 0 and 360 (decimals are allowed). This essentially rotates the Pie Chart. Note that if a 3D effect is added to the chart, the only accepted values are 90 or 270 degrees. (Equivalent to the `startAngle` property; see: startAngle.)

- **3D Group**: Creates a 3D effect if both settings in this group are higher than 0.
  - **Depth**: The depth of the Pie. (Equivalent to the `depth3D` property; see: depth3D.)
  - **Angle**: The Pie lean angle. This must be a value between 0 - 90. (Equivalent to the `angle` property; see: angle.)

Labels tab

Settings on the Label tab correspond to properties of the `AmPieChart` class in the amCharts library; see: AmPieChart.

- **Labels Group**: Defines how the label text is shown.
  - **Text**: Enter the text to use to display labels. The default text is [[title]]: [[percents]]%. Variables can be used to display specific data, <br> can be used to create a new line:
    - [[title]] : Refers to the field label.
    - [[percents]] : Contains the percentage of the Pie chart the value represents.
    - [[value]] : Contains the numerical value of the field.
    - Any text: Adding text (such as a dollar sign or column, etc) will make it appear in each label.

    (Equivalent to the `labelText` property; see: labelText.)
  - **Radius**: The distance between the label and the slice, in pixels. You can use negative values to put the label on the slice. (Equivalent to the `labelRadius` property; see: labelRadius.)
- **Tick Group**: Defines how ticks (line between the Pie chart and its labels) is shown.
  - **Color**: Enter a valid HTML hexadecimal color value (HTML Hex Color) or a predefined CSS color (CSS color names) for the color of the tick. (Equivalent to the `labelTickColor` property; see: labelTickColor.)

- **Opacity**: Enter a percentage of opacity for the tick to be displayed. The default is 20 (20% opacity). 100 is fully opaque, 0 is transparent. (Equivalent to the `labelTickAlpha` property; see: labelTickAlpha.)

- **Grouping Group**: Defines how smaller percentages are grouped together into an individual "Other" category.
  - **Less than %**: If there is more than one slice whose percentage of the pie is less than this number, those slices will be grouped together into one slice. This is the "other" slice. It will always be the last slice in a pie. (Equivalent to the `groupPercent` property; see: groupPercent.)
  - **Slice Title**: Enter a name for the label of the "Other" category. (Equivalent to the `groupedTitle` property; see: groupedTitle.)
  - **Color**: Enter a color for the slice: a valid HTML hexadecimal color value (HTML Hex Color) or a predefined CSS color (CSS color names), or click the colored square to open the Color Picker dialog ("Color Picker" on page 492). (Equivalent to the `groupedColor` property; see: groupedColor.)

Legend tab

Settings on the Legend tab correspond to properties of the AmLegend class in the amCharts library; see: AmLegend.

- **Show Legend**: Specifies if legend is enabled or not. (Equivalent to the `enabled` property; see: enabled.)
- **Legend Group**: Defines how the legends are shown.
  - **Equal label widths**: Check so that all labels are of equal width in the Legends box. The Legend's width will accommodate the largest value. (Equivalent to the `equalWidths` property; see: equalWidths.)
  - **Position**: Use the drop-down to select where the legend is shown: at the Right, Left, Top or Bottom. (Equivalent to the `position` property; see: position.)
  - **Align**: Use the drop-down to select how to align the text in the labels: Left, Middle or Right. (Equivalent to the `align` property; see: align.)

Horizontal Space: Horizontal space between legend items, in pixels. (Equivalent to the

- `spacing`
  property; see: spacing.)

Vertical Space: Enter a numerical value (in pixels) to define the vertical space between legend items, and also between the legend border and the first and last legend item. (Equivalent to the

- `verticalGap`
  property; see: verticalGap.)

Max Columns: Enter a numerical value to define the maximum number of columns in the legend. If the Legend's position is set to "right" or "left", this is automatically set to 1. (Equivalent to the

- `maxColumns`
  property; see: maxColumns.)

- **Labels Group**: Defines if and how labels are shown in the Legend.

  Text: Enter the text used to display the labels;

  - `[[title]]`
    is a variable that will be replaced with the title of the graph. (Equivalent to the `labelText`
    property; see: labelText.)

- **Markers Group**: Defines how the Legend's Markers look. Markers are icons with a color matching the Legend with its corresponding line.

  Type: Use the drop-down to select in which shape the Markers are displayed; "none" hides the Markers completely. (Equivalent to the

  - `markerType`
    property; see: markerType.)

  Size: Enter the size (in pixels) for the Markers to be displayed. (Equivalent to the

  - `markerSize`
    property; see: markerSize.)

  Label Gap: Enter the distance (in pixels) between the legend marker and legend text. (Equivalent to the

  - `markerLabelGap`
    property; see: markerLabelGap.)

  Border Width: Use the drop-down to define the thickness of the border added to the Markers. The default value (0) means the line will be a "hairline" (1 px). In case the Marker type is line, this style will be used for the line thickness. (Equivalent to the

  - `markerBorderThickness`

property; see: markerBorderThickness.)

Border Color: Color of the Legend's border. Enter a valid hexademical color (HTML Hex Color) or a predefined CSS color (CSS color names), or click the colored square to open the Color Picker dialog ("Color Picker" on page 492). (Equivalent to the

- `markerBorderColor`
  property; see: markerBorderColor.)

  Border Opacity: Enter a numerical value between 0 and 100 to define the opacity (in percentage) of the border. (Equivalent to the
- `markerBorderAlpha`
  property; see: markerBorderAlpha. When specified on the Source tab, the value should be between 0 and 1, e.g. 0.8.)

Source tab

The JSON on the Source tab reflects the choices made in the other tabs and, more importantly, provides the possibility to add in any amCharts configuration option that is unavailable via the other tab menus. For more information see: "Adding and editing properties manually" on page 265.

Print Job Summary dialog

The Print Job Summary dialog reports statistics of a print job. The entries in this summary are explained below.
If you are looking to improve the performance, please refer to the following topic: "Performance Considerations" on page 23.

- **Job**: The internal name of the template and data file that were used to create this job.
- **Statistics**: The statistics group shows all information with regards to how long the job took, the speed at which it ran, the job size and what was used to achieve those numbers.

  - **Start time**: The moment the print command was given. Typically this occurs just before the progress dialog appears.
  - **Finish time**: The moment all output files had been either written to disk or spooled to the printer. This is not the printer's finishing time.
  - **Duration**: The total time that it took to finish the job (the difference between Finish and Start time). This should be the sum of the time spent on Content Creation, Job Creation and Output Creation.

- **Content Creation**: The time spent on creating content items. This includes the time needed to process the input data and merge them with the given template.
- **Job Creation**: The time spent on assembling the job. Job creation settings can be made in the Advanced Printer Wizard, or in a separate Job Creation Preset (see "Job Creation Presets Wizard" on page 684).
- **Output Creation**: The time spent on creating the actual (spool) file(s). Output creation settings can be made in the Advanced Printer Wizard, or in a separate Output Creation Preset (see "Output Creation Presets Wizard" on page 700).
- **Input size**: The number of records in the input data.
- **Document count**: The number of documents in the job. This may be less than the input size if filtering was used in the Job Creation settings.
- **Job size**: The number of pages created during Output Creation. This number is also used to calculate the output speed in pages per minute.
- **Merge engines**: The number of Merge engines used for Content Creation. On a multi-core system it is recommended to run several Merge engines in parallel; see "Engines preferences" on page 84.
- **Cores**: The number of cores in the CPU(s) of the system.
- **Licensed speed**: The maximum speed (in pages per minute) allowed under the applicable license.
- **Speed**: The actual speed achieved (in pages per minute) in this print job. The speed is calculated by dividing Job size by Duration.

- **Message**:
  - When no message is shown, this is because the job was too short to make any sensible statement about it. This happens when the duration is below 30 seconds.
  - "Your job ran at the licensed speed." When this message is shown, the achieved speed for this job is within a 5% margin of the licensed speed. To increase the licensed speed, look into the different upgrading options (performance packs, or a product upgrade).
  - "Your job did not run at the licensed speed. Press the help button for tips on improving performance." When this message is shown, it was not possible to achieve the licensed speed. If performance is an issue, please refer to the topic about improving performance: "Performance Considerations" on page 23.

Profile Scripts dialog

The Script Profiler is accessible through the **Context > Profile Scripts** menu option. It runs the scripts in the template, using the current record, in order to verify the speed at which scripts in the "Scripts pane" on page 590 execute. It helps greatly in troubleshooting performance issues caused by scripting (see also: "Testing scripts" on page 403).

When the dialog opens, the script profiler runs automatically, on 1000 instances of all the scripts by default (this can be changed through the "Scripting preferences" on page 392).

> **Note**
>
> The script profiler can take a while, so please be patient.

The results are shown as follows (the first in the line is indicated as **Total** and represents the totals of all the scripts underneath, representing a good overview of the scripts performance in the template):

- **Name**: The name of the script being executed.
- **Count**: As the profiler runs, Count shows the current number of iterations that have been run. This goes up to the total number of set instances and then stops. Hover with your mouse to display a tooltip indicating in which sections the scripts has run (and in which contexts).
- **Elapsed**: Displays the total elapsed time since the start of the session. The table entries are initially sorted based on the values in this column, from high to low. Hovering the mouse over it will display a tooltip that indicates the breakdown of the execution time across different execution stages.
- **Delta**: Displays the estimated difference in performance between the current session and the previous session. Uses average values, so should still work if the previous session was stopped after a different number of iterations. Will be empty if no previous data is available. Hover with your mouse to display a tooltip indicating the breakdown of the execution time across different execution stages.

Rasterization Options

Before creating Email or Web output, you may want to **rasterize** certain elements, such as business graphics, to ensure that the output will be visible to most viewers. Rasterizing an

element means converting it into a JPG or PNG image. For a JPG image you can set the quality of the resulting image in a percentage.

Note

Rasterization options are only available for Boxes (elements); see "Boxes" on page 256.

Note

A business graphic in an Email section is rasterized by default and output as PNG image, because email clients usually don't support SVG images.
SVG images in an Email section give an error in the Preflight window (see "Doing a Preflight" on page 405).

See also:

- "Before generating Email output" on page 1018
- Before generating Web output

Script Debugger

The Script Debugger allows you to test personalization scripts by setting breakpoints and stepping through the scripts. (See also: "Testing scripts" on page 403.)

There are two ways to start the Script Debugger:

- Click the **Debug Scripts** button in the toolbar of the Scripts pane. The Script Debugger will pause and accept input as soon as it processes the first script.
- Right-click an enabled script in the Scripts panel and choose **Debug** from the contextual menu. The Script Debugger will pause and accept input as soon as it processes the selected script.

The Debugger simulates an output run with only the current record and appears as soon as the merge engine processes the selected script, or the first script if no script was selected. You can then add breakpoints and/or step through the code.
Meanwhile, the Workspace will display the partially merged document.

Scripts and call stack

The left side of the window displays all enabled scripts that apply to the current context and the name of the script that is displayed in the source editor at the top right and the current line number.

If the selector of a standard script or post pagination script has no matches in the current resource (Master Page or section) the script will be skipped.
If an error is thrown, the rest of the current script is skipped and execution continues with the next script.

In the call stack, frames are ordered from new to old, top to bottom. The current stack frame has an arrow indicator. You will only see multiple stack frames if execution is suspended inside a function.
You can click a frame to make it the current frame. The Variables panel always reflects the state of the current frame. Note that this does not move the instruction pointer. The script remains suspended at the most recent frame, even if that frame is not the current frame.

Source code

You can click any script at the left to view it in the source editor at the top right. An instruction pointer in the left margin marks the current line.
The code is read-only, but clicking to the left of a line of code adds a breakpoint to it (except when a line has no debuggable content, for example if it only contains a declaration or comment). Note that breakpoints are discarded when the dialog is closed.

Hovering over a variable in the code shows the value of that variable. If the variable is nested, like fields in record.fields, you can click on the variable, its parent, child elements and/or siblings (if any) to see their value.
The value is read-only, but you can select it and copy it with **Ctrl+C**.

Find code

To search for a text in all of the scripts, starting with the currently selected script, press **Ctrl+F** and start typing the text. While the Find field is visible, you can either use the arrow buttons next to it, or press **Enter** / **Shift+Enter** to go to the next or previous match, respectively.

When the Find field is not visible, you can press **F3** / **Shift+F3** to go to the next or previous code that matches your last search text.

Tip

When you click on an identifier (such as a variable name) in a script, the overview ruler at the right highlights all other occurrences of that identifier straightaway.

Variables

At the bottom right the Script Debugger shows a hierarchical overview of all variables in the scope chain and their state. This information is always relative to the current stack frame, which is the one that's selected on the left.
The overview includes the special entry "(this)", which represents the JavaScript "this" object. If the current script is not a Control Script, the overview also includes the special entry "(resource)". When expanded, this entry shows the HTML of the current Master Page or section, allowing you to see exactly how each line in the script affects the DOM.

Use the **Copy** button (or press Ctrl+C) to copy the label and contents of the selected variable and all of its children to the clipboard.

Expressions

You can add custom expressions at the top of the list of variables. This is especially useful when you want to monitor the value of a variable that is situated at a lower level in the hierarchy. For example, if you'd want to directly see the height of the top margin of a section as you step through the code, you could add the following expression: `merge.section.margins.top`.
Expressions can include function calls. For example, you could monitor the height of an element that has the ID my_div with the expression: `query("#my_div").height()`. Note that the query in this example will be limited to the current resource (Master Page or section).
You could also change the value of a variable through an expression to see how that affects the document. Let's suppose execution is suspended before an if-statement: `if (foo == 3)`. If foo is not equal to 3, but you'd still like to see what would happen if it were, you could add the expression `foo = 3`.
Expressions are evaluated automatically when execution is suspended.

- The **Add** button lets you add an expression to the overview. Expressions must be unique and cannot be empty; empty expressions will be removed.
- To **edit** an existing expression, just click the entry.
- The **Delete** button removes the selected expression.

Step-through buttons

The buttons at the bottom of the Script Debugger let you step through the code. When you step through the code it always steps relative to the most recent stack frame.

- **Into** (F5): Step into the function on the current line, if possible. A common use case for this is to step into a `results.each()` call. Note that you can only step into functions that are defined in user scripts. For other statements this behaves like (Step) Over.
- **Over** (F6): Step over the current line.
- **Return**: Step out of the current function. This only works if execution is suspended inside a function defined in one of the user scripts. Otherwise it behaves like (Step) Over.
- **Resume** (F8): Continue up to the next breakpoint or until the output run is finished.
- **Restart**: Start the debugging session over from the beginning. Execution will be suspended on the first line of the output run, even if that line does not have a breakpoint.
- **Close**: Close the Script Debugger.

Script wizards

Wizard types

Script wizards are simplified interfaces for common scripts in templates. These are the available Script wizard types.

- **Text Script**: This is the default script that is created when a data field is dragged from the Data Model onto the page. See "Variable Data" on page 351.
- **Dynamic Image Script**: Provided that its selector refers to an image, this script dynamically changes the image for each record. See "Dynamic images" on page 366.
- **Dynamic Background Script**: This type of script dynamically swaps a Print section's background. See "Dynamic Print section backgrounds" on page 368.
- **Email Scripts**. The Email To Script is automatically added to any new Email context; it defines where the email should be sent for each record. Other Email scripts define other recipients, the subject of the email that is sent, and the PDF password. See "Email header settings" on page 180.
- **Barcode Script**: This script controls the contents of a Barcode. It is automatically added when a barcode is added to a template. See "Barcode" on page 197
- **Business Graphic Script**: This script controls the contents of a Pie Chart, Bar Chart or Line Chart. See "Business graphics" on page 260.

- **Conditional script**: This script shows or hides content elements depending on certain conditions and values. See "Showing content conditionally" on page 358.

The result of the script can be either text appearing on the page, an email address or subject, the barcode data, or a JSON string that is written to the attribute of an HTML element.

Options

For the options in Conditional Script wizards see "Conditional Content script dialog" on page 539.

In other Script wizards, the following options are available.

- **Name**: The name of the script, making it easier to identify it.
- **Selector/Find text**: The Selector or Text to apply the result of the script to.
  - **Selector**: Uses CSS selectors to find the element to which the script applies
  - **Text**: Uses text as a trigger for the script. The script applies to all instances of the text found in the template.
  - **Selector and Text**: Uses text as a trigger for the script but only applies to text within the specified Selector.

> **Note**
>
> Email script wizards don't have the Selector or Find text option. Instead they select the header of the specified **Section** (the current one, by default) or **All email sections**. Note that the Section name is case-sensitive.
>
> To see the actual (HTML) selector that an email script uses to select a section, you can click the Expand button.

- **Wizard Results**: Displays a list of the data that replaces the content that matches the script's selector:
  - **Prefix**: Static text to use before the set field. For example in Dynamic Image scripts, the default prefix is **images/**.
  - **Field**: A drop-down to select which field contents to use in the script. The field should contain a valid value. For an email script, for example, the field would have to contain an email address.

    - **Format**: A special formatting modifier applied to the Field; see "Formatting variable data" on page 357.
    - **Suffix**: Static text to use after the set field. For Dynamic Image Scripts, the default suffix is **.jpg**and refers to the file extension.
- **[+]**: Adds a new line to the Wizard Results. Note that by default there is no line return between fields in the list. Adding <br> in the Suffix or Prefix field can establish a line return.
- **[-]**: Removes the currently selected line in the Wizard Results list.
- **Arrow Up**: Moves the currently selected line up one position.
- **Arrow Down**: Moves the currently selected line down one position.
- **Refresh Types**: Updates field types to match the types in the active data model.
- **Options** (only available in the Text Script wizard and the Dynamic Image wizard): specifies where and how the script inserts its results:
    - As **HTML**. HTML elements in the results are processed and displayed as HTML elements. For instance, `<b>this is bold</b>` will be displayed as **this is bold**. This is the default setting.
    - As **text**. This inserts the results as-is, meaning HTML tags and elements are displayed as text in the output. In this scenario, "<br>" shows up in the text and does not insert a line break.
    - As the value of an **attribute** of an HTML element. The selector of the script should be an HTML element. Which attributes are available depends on the selected HTML element. If the script's selector is an image (<img> element) for example, and the attribute is `src`, the script will modify the image's source. The script's results should be a valid value for the chosen attribute.
    - When checked, the option **Convert fields to JSON string** writes the results from the script into an attribute or text as a JSON string. This is useful for Web contexts where a front-end script can read this value easily.
- **OK**: Click to save any changes made to the script, apply the changes in the template, and close the dialog.
- **Cancel**: Click to close the dialog without saving changes.
- **Expand**: Click to convert the script generator to a regular script. Note that this action is not reversible once the regular script has been saved. The expanded script will use either double or single quotation marks, depending on the Preferences (**Window > Preferences > Scripting**; see "Scripting preferences" on page 392).

- **Apply**: Saves changes made to the script and applies the changes in the template without closing the dialog.

Expanded Script window

When expanded, the Script window replaces all parts of the wizard below the Selector by a box in which the script can be typed. See "Writing your own scripts" on page 393.

For an overview of keyboard shurtcuts that can be used in this script editor, see "Keyboard shortcuts" on page 561.

Chart Script dialog

The Chart Script dialog specifies which variable data will be displayed in a chart and shows a preview of both the chart (on the right hand side) and the data (at the bottom). This dialog is one of the parts of the Insert Chart wizards. It can be reopened by double-clicking the relevant script in the Scripts pane.

For more information about charts, see: "Business graphics" on page 260.

These are the options in the Chart Script dialog:

- **Name**: The name of the script, making it easier to identify it.
- **Selector**: The Selector or Text to apply the script to.
  - **Selector**: Uses CSS selectors to find the element to which the script applies. (See also: "Selectors in Connect" on page 414.)
  - **Text**: Uses text as a trigger for the script. The script applies to all instances of the text found in the template.
  - **Selector and Text**: Uses text within the specified Selector as a trigger for the script.
- **Input Type**: Use the drop-down to select the source of the data to add to the Chart.
  - **Data fields**: Select to use fields at the record level in the Data Model.
  - **Data table**: Select to use fields in a detail table. At least one detail table must be available in the Data Model Pane for this option to be functional.
    - **Detail table**: Use the drop-down to select one of the detail tables that are part of the Data Model.

- **Category**: Use the drop-down list to select the data field (in the selected detail table) of which the values will appear under the bars or the line; in other words, on the **x** axis.

- **Values**: Select data fields with a numerical value.
  - **Column**: Removing the topmost check deselects all data fields.
  - **Labels**: Click the label to enter a new label for a field. It depends on the Input Type where the labels appear: with Data Fields, the labels appear under the x-axis of the chart; with a Data Table, labels appear in the Legend.
  - **Color**: Click on the color to open the Color Picker dialog ("Color Picker" on page 492) and select a different color.

  Buttons next to Values:
  - The **Toggle non-numeric fields** button filters non-numeric fields from the list. The list will then display only Integer, Float and Currency data fields.
  - The **Move Up** and **Move Down** buttons change the order of the fields, which is reflected in the chart.
- **OK**: Click to save any changes made to the script, apply the changes in the template, and close the dialog.
- **Cancel**: Click to close the dialog without saving changes.
- **Expand**: Click to convert the script generator to a regular script. Note that this action is not reversible once the regular script has been saved.
  The expanded script will use either double or single quotation marks, depending on the Preferences (**Window > Preferences > Scripting**; see "Scripting preferences" on page 392).
- **Apply**: Saves changes made to the script and applies the changes in the template without closing the dialog. Note that the effect will only be visible on the Preview tab in the Workspace.

Expanded Script window

When expanded, the Script window replaces all parts of the wizard below the Selector by a box in which the script can be typed. See "Writing your own scripts" on page 393.

For an overview of keyboard shortcuts that can be used in this script editor, see "Keyboard shortcuts" on page 561.

Conditional Content script dialog

Conditional Content scripts can show or hide elements depending on certain conditions and values. They can be added by right-clicking any element in a template and clicking **Make Conditional**. If the current element has neither an ID nor a class, an ID will be automatically generated. See "Showing content conditionally" on page 358.

The general options in the Conditional Script wizard are:

- **Name**: The name of the script, making it easier to identify it.
- **Selector**: The Selector or Text to apply the result of the script to.
  - **Selector**: Uses CSS selectors to find the element to which the script applies.
  - **Text**: Uses text as a trigger for the script. The script applies to all instances of the text found in the template.
  - **Selector and Text**: Uses the specified selector as a trigger for the script. The script applies to all instances of the text found within the specified Selector.

  For more information about Selectors see "Selectors in Connect" on page 414.
- **Action**: Use the drop-down to select whether to **Show** or **Hide** the element when the condition below is true. If, conversely, the condition evaluates to false, the opposite action will be performed.

A **condition** is made up of groups and rules.

The **Add** button adds a rule that evaluates a **data field** to a group.
To add either a **group**, or a rule that evaluates a **runtime parameter**, click the downward pointing arrow next to this button and select **Group** or **Parameter Rule**.

- **Group**:
  A group consists of one or more rules with a logic operator. Four logic choices are available at the Group level. These are:
  - **All of the following**.
    This equates to the logical operator (... AND ...).
    If *all* of the associated criteria are met, then this group resolves to TRUE.
  - **Any of the following**.
    This equates to the logical operator (... OR ...).
    If *any* of the associated criteria are met, then this group resolves to TRUE.
  - **Not all of the following**.
    This equates to the logical operator (NOT(... AND ...)).

If *any (but not all)* of the associated criteria are met, then this group resolves to TRUE.

- **Not any of the following**.
  This equates to the logical operator (NOT (... OR ...)).
  If *none* of the associated criteria are met, then this group resolves to TRUE.

- **Rule**:
  - **Data Field** / **Parameter**: Use the drop-down to select which data field in the record, or which runtime parameter in the template, the condition will be based upon. (See "Variable Data" on page 351 and Runtime parameters.)
  - **Operator**: Select which kind of comparison is applied. The options are: *Equal to*, *Not equal to*, *Contains*, *Does not contain*, *Begins with*, *Ends with*.
  - **Value**: The value(s) used for the conditional check. Values are **case sensitive** by default. You can click the button next to the value to make them case insensitive, which means that upper- and lowercase letters are treated as being the same.

The **script** is displayed below the condition and updated while the condition is being edited.

Section properties dialogs

The respective section properties dialogs are opened via the contextual menu:

- Right-click the section in the **Resources** pane and choose **Properties**.

Email section properties

For information about Email sections, see: "Email templates" on page 177.

Properties tab

The properties for an Email section are minimal and contain the following options:

- **Name**: Enter the name of the Section in the Email Context. This has no effect on output.
- **CSS mode**: In the Designer, CSS files can be used to style email templates (see "Styling templates with CSS files" on page 296) but eventually, the formatting must be applied in a different way since email clients do not read CSS files. This property determines if and how the styles in linked CSS files are applied to the output.
  - **Write CSS to <head> section**: The content of linked style sheet files is copied to <style> elements in the <head> of the HTML. (This is also known as "embedded

CSS".) With this option email gets sent faster; however, email clients may not support <style> elements in the <head> and the order of the CSS rules could be different.
This is the recommended setting, and the default setting for new templates.

- **Apply CSS properties to elements**: The relevant CSS properties are copied to the <style> attribute of the various HTML elements.
  This is the default setting for templates made with PrintShopMail Connect versions prior to 2020.1.
- **Do not inline styles**: No styles are copied from linked CSS style sheets. It is assumed that you format elements locally or use a CSS inliner tool.

> **Note**
>
> It can't be guaranteed that content will look the same in all email clients, particularly when it comes to older client versions, so it is recommended to always test and validate the output. See: "Testing Email output" on page 1020.

- The **Meta Information** group lists all <meta> tags that will be added to the header of the email. See "Meta information" on page 186.
- When the option **Append plain-text copy of the HTML** is checked, a plain-text version of the HTML is added to each email that is sent. With new templates this option is checked by default.

Includes tab

This tab lists the style sheets that can be applied to the email section when producing the output. Style sheets are loaded in the order shown, and styles in later style sheets overwrite earlier ones when the same selector is used. (See "Includes dialog" on page 504 and "Styling templates with CSS files" on page 296.)

Attachments tab

The **Attachments** tab lets you select files and delete attachments. For more information, see: "Email attachments" on page 187.

Print section properties

For information about Print sections, see: "Print sections" on page 137.

Properties

- **Section** group:
  - **Name**: The name of the section.
  - **Minimum number of pages**: The default minimum number of pages is 1. If content is too long to fit on one page, overflow will automatically appear on the next page. Changing the minimum number of pages can be useful when you want certain content to appear on a specific page.

  These properties can be used in a Control Script; see the topic "Control Scripts" on page 429 and the page about the section object: "section" on page 978.

For an explanation of other settings on the Properties tab, see "Page settings: size, margins and bleed" on page 148.

Includes Tab

The Includes tab defines which style sheets and JavaScript files should be applied to the section when generating output, and in which order; see "Includes dialog" on page 504.
For more information about stylesheets, see "Styling templates with CSS files" on page 296.
For more information about using JavaScript, see Using JavaScript.

Finishing tab

This tab defines finishing options for this section when it is printed. For an explanation of all Binding and Hole making options, see "Finishing options" on page 691.

Sheet Configuration tab

The Sheet Configuration tab defines how different Print context sections output on different Media (see "Media" on page 159) and using different Master Pages (see "Master Pages" on page 156). For an explanation of all settings on this tab, see "Sheet Configuration dialog" on page 546.

Background tab

This tab defines the background of the current Print section; see "Using a PDF file or other image as background" on page 142.

Numbering tab

The Numbering tab defines how page numbers are configured in the current Print section; see "Configuring page numbers" on page 152.

Arrange Sections

The Arrange dialog is used to change the order of sections within a context. This can have an effect on how they are outputted. See: "Print sections" on page 137 and "Email templates" on page 177.

To access the Arrange dialog, on the Resources pane, right-click on any section or the context containing them, and click **Arrange**.

- **Name**: Displays the name of each section within the context.
- **Move Up**: Click to move the currently selected section up one position.
- **Move Down**: Click to move the currently selected section down one position.

Send (Test) Email

The Send Email dialog is used to generate mail output and send it to each recipient in the Record Set. To open this dialog, select **File > Send Email**, on the menu.
Note that the subject, recipients etc. must be specified before sending the email; see "Email header settings" on page 180.
For more information about the process of sending out email and the possible settings, see "Generating Email output" on page 1017.

Options for this dialog:

- **From** group:
  - **Name**: Enter the name that should appear when sending emails. The name is optional.
  - **Email**: Enter the email address that will appear as a Sender to the email recipient. A single email address should be written.
- **To** group - only when sending a test email:
  - **Email address(es)**: Enter one or more email addresses where the test emails are sent. Multiple emails can be separated by semicolons (;), and can be in the same format as above. Note that every email here will receive all the emails for the record-range below.

- **Use Litmus**: Check to also send the emails to the Litmus test email set in the Email Preferences (to go to the Email Preferences, select **Window > Preferences**, click the arrow next to **Email**, and then click **General**). Disabled if no Litmus email is set.

- **Records** group:
  - Select **All**, or click **Selection** and enter the range of records that should be sent. Removing the range disables the selection and sends emails to all records in the record set.
- **Attachments**:
  - **Print context as PDF**: If a Print Context exists in the template, its output will be generated and a PDF version of it will be attached to the outgoing email.
- **Outgoing mail settings:**
  - **Presets**: Use the drop-down to select a preset. These presets are configured in the Email (SMTP) preferences; see "Email SMTP settings" on page 182.

> **Note**
>
> It is recommended to use an Email Service Provider to get access to tools that give you full control over your mailings, like open rates, click through rates etc. See "Using an ESP with PrintShop Mail Connect" on page 1021.

  - **Host**: The SMTP server through which the emails are to be sent. Can be a host (mail.domain.com) or an IP address. The port number can be appended to the host name, separated by a colon, for example: smtp.mandrillapp.com:465. Port 25 will be used if no port number is specified.

> **Note**
>
> If the mail server supports it, the connection will be encrypted without the need to send the server a STARTTLS instruction when port 465 is used.

> **Tip**
>
> Gmail only allows Connect to be used as an SMTP client when "Access for less secure apps" is enabled in the Google account settings.

- **Use authentication**: Check if a user name and password are needed to send emails through the host.
- **Send STARTTLS**: Enabled if authentication is checked. With STARTTLS the client negotiates with the mail server to use some form of encryption, usually a version of Transport Layer Security (TLS). Since this improves security it is recommend to enable this option if you use port 25 (the default port), 2525, or 587.
  Note that the email will not be sent if the SMTP server does not support TLS or SSL (an older encryption type).
  This option is ignored when port 465 is used.
- **User**: Enter the user name used to connect to the SMTP server.
- **Password**: Enter the password for the above user name.

Select Image dialog

The Select Image dialog lets you select an image, depending on where the image is located. Whenever possible, a preview of the selected image will be shown.

- Click **Resources**, **Disk** or **Url**, depending on where the image is located.
  - **Resources** lists the images that are present in the **Images** folder on the **Resources** pane.
  - **Disk** lists image files that reside in a folder on a hard drive that is accessible from your computer. Click the **Browse** button to select a folder (or an image in a folder). As an alternative it is possible to enter the path manually. You can give a local path (e.g. C:\Images\Test.jpg) or use the "file" protocol. The complete syntax of a fully qualified URL with the "file" protocol is: `file://<host>/<path>`. Note: if the host is "localhost", it can be omitted, resulting in `file:///<path>`, for example: `file:///c:/resources/images/image.jpg`.
  - **Url** lists image files from a specific web address. Select the protocol (**http** or **https**), and then enter a web address (for example, http://www.mysite.com/images/image.jpg).

Note

If a URL doesn't have a file extension, and the option Save with template is **not** selected, the Select Image dialog automatically adds the `filetype` parameter with the file extension as its value (for example: `?filetype=pdf` (if it is the first parameter) or `&filetype=pdf`).
The `filetype`, `page` and `nopreview` parameters are not sent to the host; they are used internally. Therefore, URLs that rely on one of these parameters cannot be used.

- With an external image, you can check the option **Save with template**. If this option is checked, the file will be inserted in the **Images** folder on the **Resources** pane at the top left.
  If it isn't saved with the template, the image remains external. Note that external images need to be available when the template is merged with a record set to generate output, and that their location should be accessible from the machine on which the template's output is produced. External images are updated (retrieved) at the time the output is generated.

Sheet Configuration dialog

The Sheet Configuration dialog defines how different Print context sections output on different Media (see "Media" on page 159) and using different Master Pages (see "Master Pages" on page 156).

Note

Master Pages, Media and Duplex printing options can also be set in a Control Script (see "Control Scripts" on page 429 and "Control Script API" on page 931). This is especially useful when you need identical sections with different settings.

General options

The first option defines **Duplex** printing, which also enables or disables the settings for the **Back** side of each sheet.
If Duplex is enabled, you can also:

- Check **Omit empty back side for Last or Single sheet** to reset a page to Simplex if it has an empty back side (no content and no Master Page). Thus changing a Duplex job into a Mixplex job may reduce volume printing costs as omitted back sides aren't included in the number of printed pages. On the other hand, depending on the printer type it may reduce the print speed as well.
- Check **Tumble** to duplex pages like in a calendar. (On Portrait output, this would be equivalent to short-edge duplex.)
- Check **Facing pages** to have the margins of the section switch alternately, so that pages are printed as if in a magazine or book.

The **Media rotation** setting rotates the Media 0, 90, 180, or -90 degrees.

Sheet position options

With the option **Same for all positions** checked, the same Master Page and Media will be applied to every page in the Print section.
When this option isn't checked, there are multiple groups, each defining the settings for pages grouped by their position within the section as it outputs: **First**, **Middle**, **Last** and **Single** sheets.

Each group defines:

- **Allow content on**: Selects on which face of the sheet content is allowed.
  If **Front only** or **Back only** is selected, the page acts as a Simplex page even though Duplex printing is enabled. The other page may contain a Master Page, but no contents will be printed on it. As such it does not count in the Content Page Number and Content Page Count markers which can be inserted via the Insert menu (see "Page numbers " on page 150).
- **Media**: Defines the media that is used. If the Media has Virtual Stationery defined, the selected image is shown as a background to each page that corresponds to the media's sheet position.
  - **Edit Script**: Click this button to create a script that defines which Media is used.
- **Master Page Front**: Defines the Master Page used for the front of the selected sheet's position. (Disabled if Back only is selected under Allow content on).
- **Master Page Back**: Defines the Master Page used for the back of the selected sheet's position. (Disabled if Front only is selected under Allow content on, or if Duplex is unchecked.)
  Omit Master Page Back in case of an empty back page omits the specified Master Page on the last backside of a section if that page is empty. That page will then also be

skipped from the page count *unless* the page numbers continue on the next section (see "Configuring page numbers" on page 152).
Note that if the **Omit empty back side for Last or Single sheet** option (see "General options" on the previous page) is checked as well, the empty backside will not appear in the output at all and will not be counted in any case.

The **Repeat sheet configuration** option can be used to have the sheet configuration repeat every *n* number of pages. This is useful when documents can be so long that they cannot fit into one envelope.

Style sheets dialog

The Stylesheet editor dialog is used to edit CSS style sheet resources. For information on the use of style sheets, see "Styling and formatting" on page 294 and "Styling templates with CSS files" on page 296.

This dialog lets you edit the Global style sheet (context_all_styles.css, which by default applies to all contexts), and the style sheet that applies to the context that is currently being edited in the workspace: Print (context_print_style.css) or Email (context_email_style.css).

To open this dialog, select **Edit > Stylesheets...**.

- **Context**: Use the drop-down to select **Global** (all contexts) or the context that is open in the workspace, such as **Print**. Selecting a context shows all it's CSS rules in the **Rule List**.
- **Show**: Use the drop-down to select whether to show all CSS rules or limit to certain types: **Class**, **ID** or **Element** rules.
- **Rule List**: Displays the list of rules in the currently selected style sheet.
- **Rule Display**: Displays the contents of the currently selected rule in the **Rule List**.
- **New**: Click to create a new rule with the Edit Rule dialog. See "New/Edit Rule dialog" on the next page.
- **Edit**: Click to edit the currently selected rule in the Rule List using the Edit Rule dialog. See "New/Edit Rule dialog" on the next page.
- **Delete**: Click to delete the currently selected rule in the Rule List.
- **Duplicate**: Click to create a duplicate of the currently selected rule in the Rule List using the Edit Rule dialog. The default name for the new rule is the name of the current one plus "-duplicated". See "New/Edit Rule dialog" below.
- **Move Up**: Move the currently selected rule in the Rule List up one position in the list.

- **Move Down**: Move the currently selected rule in the Rule List down one position in the list.
- **Save**: Click to save all changes to the style sheet and close the dialog.
- **Cancel**: Click to close the dialog without saving any changes.

New/Edit Rule dialog

The New/Edit Rule dialog shows the properties for a specific CSS selector.
Click the **Apply** button to see how a setting affects the elements that are subject to that selector. (You may have to move and resize the Stylesheet dialog before opening the Edit Rule dialog, in order to be able to see the template that you are working on.)
At any point you can click on the **Advanced** button to see the Advanced Stylesheet Rule. See "Advanced Stylesheet Rule" on page 552.

- **Name**: The CSS Selector to which this rule applies. This can be a selector that is specific to Connect (see "Selectors in Connect" on page 414) or any selector used in regular CSS, since they can also be used here (see CSS Selectors on W3Schools for a simple reference page).

Type Tab

- **General group**:
  - **Font**: Select the font used to display text. See also: "Fonts" on page 332. This is equivalent to the CSS `font-family` property.
  - **Size**: Enter the size in a measure, named size or percentage. This is equivalent to the CSS `font-size` property.
  - **Color**: This is the color of the text. Select a named font color as defined in the Edit Colors dialog (see "Colors" on page 328) or click the colored square to create a new color or to enter a color manually. The color value must be a valid HTML color name or hexadecimal color code. This is equivalent to the CSS `color` property.
  - **Background Color**: This is the background color of the text. Select a named font color as defined in the Edit Colors dialog (see "Colors" on page 328) or click the colored square to create a new color or to manually enter a color value (a valid HTML color name or hexadecimal color code). This is equivalent to the CSS `background-color` property.
- **Spacing group**:
  - **Letter Spacing**: Set the space between characters in a text in measure or percentage. This is equivalent to the CSS `letter-spacing` property.

- **Word Spacing**: Set the space between each word in a text in measure or percentage. This is equivalent to the CSS `word-spacing` property.
- **Whitespace**: Specify how to handle white spaces inside of an element. See CSS White-Space for details. This is equivalent to the CSS `white-space` property.

- **Style group**: Check any option to apply the selected style to text within the element. This list shows the CSS property and value for each of the options.
  - **Bold**: Sets the `font-weight` to `700`.
  - **Italic**: Sets `font-style` to `italic`.
  - **Underline**: Sets `text-decoration` to `underline`.
  - **Strikethrough**: Sets `text-decoration` to `line-through`.
  - **Subscript**: Sets `vertical-align` to `super`.
  - **Superscript**: Sets `vertical-align` to `sub`.
  - **Capitalize**: Sets `text-transform` to `capitalize`.
  - **Uppercase**: Sets `text-transform` to `uppercase`.
  - **Lowercase**: Sets `text-transform` to `lowercase`.
  - **Small-caps**: Sets `font-variant` to `small-caps`.

Formats Tab

- **General group**:
  - **Line-height**: Specify the height of each line in the element's text, in measure or percentage. Note that this is not spacing between lines, but rather the complete height of the line itself including the text. Equivalent to the `line-height` property.
  - **Align**: Select how text should be aligned, such as `left`, `center`, `right` or `justify`. Equivalent to the `align` property.
  - **First Indent**: Specify the indentation of the first line of each paragraph in the element. Equivalent to the `text-indent` property.
  - **Display**: Select how to display the element. This can also be used to hide an element completely using the `none` option. See CSS Display. Equivalent to the `display` property.
- **Breaks group**:
  - **Before**: Specifies whether a page break should occur before the element. Equivalent to the `page-break-before` property.

- **Inside**: Specifies whether to accept page breaks within the element. Equivalent to the `page-break-inside` property.
- **After**: Specifies whether a page break should occur after the element. Equivalent to the `page-break-after` property.
- **Widows**: Specifies how to handle widows within a paragraph (lines appearing alone on the next page if the paragraph does not fit on the current one). Equivalent to the `widows` property. Widows and orphans are ignored if the `page-break-inside` property is set to `avoid`.
- **Orphans**: Specifies how to handle orphans within a paragraph (lines appearing alone at the end of a page if the paragraph does not fit on the current one). Equivalent to the `orphans` property.

Spacing Tab

See also: "Spacing" on page 337.

- **Padding group**: Defines padding (spacing inside the element) in measure or percentage:
  - **All sides**: Check to set all padding to use the Top value. Equivalent to the `border` property.
  - **Top, Left, Bottom, Right**: Set padding for each side. Equivalent to the `border-left`, `border-top`, `border-right` and `border-bottom` properties.
- **Margin group**: Defines margins (spacing outside the element) in measure or percentage:
  - **All sides**: Check to set all margins to use the Top value. Equivalent to the `margin` property.
  - **Top, Left, Bottom, Right**: Set the margin for each side. Equivalent to the `margin-left`, `margin-top`, `margin-right` and `margin-bottom` properties.

Border Tab

See also: "Border" on page 325.

- **Same for all sides**: Defines the border properties for all sides using the Top properties. Equivalent to the`border`property.
- **Top, Left, Bottom, Right**: Each group defines the following properties:
  - **Width**: Specify the thickness of the border. Equivalent to the `border-width` property.
  - **Style**: Specify the style of the border such as `solid`, `dashed` or `dotted`. Equivalent to the `border-style` property.

- **Color**: Specify the color of the border. The color value must be a valid HTML Color Name, or a valid HTML Hex Color. Equivalent to the `border-color` property.

Advanced Stylesheet Rule

The Advanced editor is used to manually input rules. Note that to use this dialog, basic knowledge of CSS rules is a pre-requisite, as no check is currently done to verify that properties and values are correct.

- Property List: Lists all the currently available properties for the selector.
  - **Property**: The name of the property. This must correspond exactly to a known property (see CSS Reference). An autocompletion drop-down displays to show possible values when typing.
  - **Value**: The value for the given property. The values must be valid for that property, see the CSS Reference link above and check the property for valid values.
- **New**: Click to create a new line and type in the property.
- **Delete**: Click to delete the currently selected property in the Property List.
- **Move Up**: Move the currently selected property in the Property List up one position in the list.
- **Move Down**: Move the currently selected property in the Property List down one position in the list.

Sync Summary dialog

The Sync Summary dialog reports any discrepancies between the elements in the template that are tagged for translation (see "Tagging elements for translation" on page 469) and the translation entries on the "Translations pane" on page 594. To open it, click the **Sync** button on the **Translations** pane.

Remote snippets and content added by scripts are not taken into account.

For more information about template translation, see "Translating templates" on page 467.

- **New Strings**: Lists the texts, found in elements that are tagged for translation, for which there is no matching translation entry.
- **Orphaned Strings**: Lists the translation entries that seem to be unused: currently there is no matching element in the content of the template for them. Note that seemingly unused translation entries may have been added for content in a remote snippet or for content that will be added by a script.

Per translation entry you have the following options.

- **Source**: This is the text that will be presented to the translator, and that will be replaced with a translation when OL Connect generates output from the template. It is case-sensitive.
- **Context** (optional): Describe where the source text is used, for example: "This text is in the head of a table", or "Table|Head", to help the translator make the best choice when the source text appears more than once and should be translated differently in different locations.
  The description is added to the <data-translate> attribute of the HTML element in which the source text is located, as its value, for example: <data-translate="Table">.
  Note that a translation will only be applied if the value of the <data-translate> attribute of the respective HTML element is *exactly* the same as the given Context (case-sensitive).

Click **OK** to update the Translations pane and add the new translation entries.

> **Tip**
>
> The **pluralization** option and the option to write **instructions for the translator** are only available in the Translation String Options dialog. After closing the Sync Summary dialog, double-click on a translation entry in the Translations pane to open it.

Table Formatting dialog

The Table Formatting dialog defines how a table looks. Note that the settings are applied to the table as a whole. For example, when you change the border of the table, the borders of cells inside the table will not be changed. For more information see "Styling a table" on page 316.

All settings in this dialog are in fact CSS properties. Cascading Style Sheets (CSS) were originally designed for use with web pages: HTML files. Since Designer templates are HTML files, they are styled with CSS. To learn how to use CSS in the Designer, see "Styling and formatting" on page 294 and "Styling templates with CSS files" on page 296. For information about specific properties and their options, see W3Schools CSS Reference.

Note

Using viewport based units (vw, vh, vmin, vmax) in Print sections is not recommended. This may cause differences between the preview in the Designer and the printed output.

Table Tab

- **General group**:
  - **Width**: Set the width of the table in measure or percentage. Equivalent to the CSS `width` property.
  - **Height**: Set the height of the table in measure or percentage. Equivalent to the CSS `height` property.
  - **Angle**: Set the rotation angle of the table in clockwise degrees. Equivalent to the CSS `transform:rotate` property.
  - **Corner radius**: Set the radius of rounded border corners in measure or percentage. Equivalent to the CSS `border-radius` property.
  - **Display**: Use the drop-down or type in the value for how to display the table. Equivalent to the CSS `display` property.
  - **Overflow**: Use the drop-down or type in the value for how to handle overflow (text that does not fit in the current size of the box). Equivalent to the CSS `overflow` property.
- **Text wrap group**:
  - **Float**: Use the drop-down or type in the value for how to float the table, if the table is not in an absolute position. Equivalent to the CSS `float` property.
  - **Clear**: Use the drop-down or type in the value for clearing pre-existing alignments. Equivalent to the CSS `clear` property.
- **Positioning**:
  - **Position**: Use the drop-down or type in the value for the type of positioning for the table. Equivalent to the CSS `position` property.
  - **Top**: Set the vertical offset between this table and its parent's top position. Equivalent to the CSS `top` property.
  - **Left**: Set the horizontal offset between this table and its parent's left position. Equivalent to the CSS `left` property.

- **Bottom**: Set the vertical offset between this table and its parent's bottom position. Equivalent to the CSS `bottom` property.
- **Right**: Set the horizontal offset between this table and its parent's left position. Equivalent to the CSS `right` property.
- **Z-index**: Set the z-index of the table. The z-index defines in which order elements appear. Equivalent to the CSS `z-index` property.

- **Breaks group**:
  - **Before**: Specifies how to handle page breaks before the table. Equivalent to the CSS `page-break-before` property.
  - **Inside**: Specifies whether to accept page breaks within the table. Equivalent to the CSS `page-break-inside` property.
  - **After**: Specifies how to handle page breaks after the table. Equivalent to the CSS `page-break-after` property.
  - **Widows**: Specifies how to handle widows within the table (rows appearing alone on the next page if the table does not fit on the current one). Equivalent to the CSS `widows` property. Widows and orphans are ignored if the `page-break-inside` property is set to `avoid`.
  - **Orphans**: Specifies how to handle orphans within the tables (rows appearing alone at the end of a page if the table does not fit on the current one). Equivalent to the CSS `orphans` property.

Spacing Tab

For information about spacing see "Spacing" on page 337.

- **Padding group**: Defines padding (spacing inside the element) in measure or percentage:
  - **All sides**: Check to set all padding to use the Top value. Equivalent to the CSS `padding` property.
  - **Top, Left, Bottom, Right**: Set padding for each side. Equivalent to the CSS `padding-left`, `padding-top`, `padding-right` and `padding-bottom` properties.
- **Margin group**: Defines margins (spacing outside the element) in measure or percentage:
  - **All sides**: Check to set all margins to use the Top value. Equivalent to the CSS `margin` property.

- **Top, Left, Bottom, Right**: Set the margin for each side. Equivalent to the CSS `margin-left`, `margin-top`, `margin-right` and `margin-bottom` properties.

Border Tab

For information about borders see "Border" on page 325.

- **Same for all sides**: Defines the border properties for all sides using the Top properties. Equivalent to the CSS `border` property.
- **Top, Left, Bottom, Right**: Each group defines the following properties:
  - **Width**: Specify the thickness of the border. Equivalent to the CSS `border-width`property.
  - **Style**: Specify the style of the border such as `solid`, `dashed` or `dotted`. Equivalent to the CSS `border-style` property.
  - **Color**: Specify the color of the border: select a named color (defined in the "Colors Properties" on page 492) from the drop-down, or click the colored square to open the Color Picker dialog ("Color Picker" on page 492). Alternatively you could type a name or value in the Color field directly. It must be a predefined CSS color name (CSS color names), a hexadecimal color code (HTML Hex Color), an RGB color value, for example `rgb (216,255,170)` or a CMYK color value, for example `cmyk(15%, 0%, 33%, 0%)`.
    Equivalent to the CSS `border-color` property.

Background Tab

For information about backgrounds see "Background color and/or image" on page 323.

- **General group**:
  - **Color**: Specify the color of the table background:select a named color (defined in the "Colors Properties" on page 492) from the drop-down, or click the colored square to open the Color Picker dialog ("Color Picker" on page 492). Alternatively you could type a name or value in the Color field directly. It must be a predefined CSS color name (CSS color names), a hexadecimal color code (HTML Hex Color), an RGB color value, for example `rgb (216,255,170)` or a CMYK color value, for example `cmyk(15%, 0%, 33%, 0%)`.
    Equivalent to the CSS `background-color` property.

- **Background image group**:
  - **Source**: Click the **Select Image** button to select an image via the "Select Image dialog" on page 545. Equivalent to the CSS `background` property.
  - **Size**: Select `auto`, `cover` or `contain` (for an explanation see https://www.w3schools.com/cssref/css3_pr_background-size.asp), or type the width and height of the image in a measure (e.g. `80px 60px`) or as a percentage of the parent element's size (e.g. `50% 50%`). Equivalent to the CSS `background-size` property.
  - **Position**: Select the position for the background-image. Equivalent to the CSS `background-position` property.

Table Cell Formatting dialog

The Table Cell Formatting dialog defines how a particular cell in a table looks. For more information see "Styling a table" on page 316.

All settings in this dialog are in fact CSS properties. Cascading Style Sheets (CSS) were originally designed for use with web pages: HTML files. Since Designer templates are HTML files, they are styled with CSS. To learn how to use CSS in the Designer, see "Styling and formatting" on page 294 and "Styling templates with CSS files" on page 296. For information about specific properties and their options, see W3Schools CSS Reference.

> **Note**
>
> Using viewport based units (vw, vh, vmin, vmax) in Print sections is not recommended. This may cause differences between the preview in the Designer and the printed output.

Cell Tab

- **Width**: Set the width of the table in measure or percentage. Equivalent to the CSS `width` property.
- **Height**: Set the height of the table in measure or percentage. Equivalent to the CSS `height` property.
- **Vertical Align**: Specify how text is vertically aligned in the cell: top, middle, bottom or baseline. With the baseline value all the table data share the same baseline. Often this has the same effect as the bottom value. However, if the fonts are in different sizes, baseline looks better.

Type Tab

- **General group**:
  - **Font**: Select the font used to display text, equivalent to the CSS `font-family` property.
  - **Size**: Enter the size in measure, named size or percentage, equivalent to the CSS `font-size` property.
  - **Color**: You may select a named color (defined in the "Colors Properties" on page 492) from the drop-down, or click the colored square to open the Color Picker dialog ("Color Picker" on page 492). Alternatively you could type a name or value in the Color field directly. It must be a predefined CSS color name (CSS color names), a hexadecimal color code (HTML Hex Color), an RGB color value, for example `rgb(216,255,170)` or a CMYK color value, for example `cmyk(15%, 0%, 33%, 0%)`.
    Equivalent to the CSS `color` property.
- **Spacing group**:
  - **Letter Spacing**: Set the space between characters in a text in measure or percentage. Equivalent to the CSS `letter-spacing` property.
  - **Word Spacing**: Set the space between each word in a text in measure or percentage. Equivalent to the CSS `word-spacing` property.
  - **Whitespace**: Specify how to handle white spaces inside of an element. Equivalent to the CSS `white-space` property. See CSS White-Space for details.
- **Style group**: Check any option to apply the selected style to text within the element:
  - **Bold**: Sets the `font-weight` to `700`.
  - **Italic**: Sets the `font-style` to `italic`.
  - **Underline**: Sets the `text-decoration` to `underline`.
  - **Strikethrough**: Sets the `text-decoration` to `line-through`.
  - **Subscript**: Sets the `vertical-align` to `super`.
  - **Superscript**: Sets the `vertical-align` to `sub`.
  - **Capitalize**: Sets the `text-transform` to `capitalize`.
  - **Uppercase**: Sets the `text-transform` to `uppercase`.
  - **Lowercase**: Sets the `text-transform` to `lowercase`.
  - **Small-caps**: Sets the `font-variant` to `small-caps`.

Spacing Tab

For information about spacing see "Spacing" on page 337.

- **Padding group**: Defines padding (spacing inside the element) in measure or percentage:
  - **All sides**: Check to set all padding to use the Top value. Equivalent to the CSS `border` property.
  - **Top, Left, Bottom, Right**: Set padding for each side. Equivalent to the CSS `padding-left`, `padding-top`, `padding-right` and `padding-bottom` properties.

Border Tab

For information about borders see "Border" on page 325.

- **Same for all sides**: Defines the border properties for all sides using the Top properties. Equivalent to the `border` property.
- **Top, Left, Bottom, Right**: Each group defines the following properties:
  - **Width**: Specify the thickness of the border. Equivalent to the CSS `border-width` property.
  - **Style**: Specify the style of the border such as `solid`, `dashed` or `dotted`. Equivalent to the CSS `border-style` property.
  - **Color**: Specify the color of the border: select a named color (defined in the "Colors Properties" on page 492) from the drop-down, or click the colored square to open the Color Picker dialog ("Color Picker" on page 492). Alternatively you could type a name or value in the Color field directly. It must be a predefined CSS color name (CSS color names), a hexadecimal color code (HTML Hex Color), an RGB color value, for example `rgb(216,255,170)` or a CMYK color value, for example `cmyk(15%, 0%, 33%, 0%)`.
    Equivalent to the CSS `border-color` property.

Background Tab

For information about backgrounds see "Background color and/or image" on page 323.

- **General group**:
  - **Color**: Specify the color of the table cell background: select a named color (defined in the "Colors Properties" on page 492) from the drop-down, or click the colored

square to open the Color Picker dialog ("Color Picker" on page 492). Alternatively you could type a name or value in the Color field directly. It must be a predefined CSS color name (CSS color names), a hexadecimal color code (HTML Hex Color), an RGB color value, for example `rgb(216,255,170)` or a CMYK color value, for example `cmyk(15%, 0%, 33%, 0%)`.
Equivalent to the CSS `background-color` property.

- **Background image group**:
  - **Source**: click the **Select Image** button to select an image via the "Select Image dialog" on page 545. Equivalent to the CSS`background`property.
  - **Size**: select `auto`, `cover` or `contain` (for an explanation see https://www.w3schools.com/cssref/css3_pr_background-size.asp), or type the width and height of the image in a measure (e.g. `80px 60px`) or as a percentage of the parent element's size (e.g. `50% 50%`). Equivalent to the CSS `background-size` property.
  - **Position**: select the position for the background-image. Equivalent to the CSS `background-position` property.

Translation String Options dialog

The Translation String Options dialog sets the options for a translation entry. To open it, double-click a translation entry or click the Add Empty String button on the **Translations** pane.

For information about template translation, see "Translating templates" on page 467.

- **String**:
  - **Source**: This is the text that will be presented to the translator, and that will be replaced with a translation when OL Connect generates output from the template. It is case-sensitive.
    When a new translation entry is created by tagging an HTML element for translation (see "Tagging elements for translation" on page 469), the text of that element gets copied to this field.
  - **Context** (optional): Describe where the source text is used, for example: "This text is in the head of a table", or "Table|Head", to help the translator make the best choice when the source text appears more than once and should be translated differently in different locations.
    The description is added to the <data-translate> attribute of the HTML element in which the source text is located, as its value, for example: <data-translate="Table">. Note that a translation will only be applied if the value of the <data-translate>

attribute of the respective HTML element is *exactly* the same as the given Context (case-sensitive).

- **Pluralization**: Check this option to make it possible for the translator to enter (text with) plurals. See "Pluralization" on page 472.

> **Note**
>
> The Singular and Plural given in this dialog are only used when there is no .po file that matches the locale.

  - **Singular**: Enter the text to use when the value of the selected data field is 1.
  - **Plural**: Enter the text to use when the value of the selected data field is **not** 1.
  - **Data field**: Select a data field that contains a numerical value. It depends on the value in this field whether the singular or plural is used.
- **Instructions for the translator**:
  - **Comment**: Enter a comment that helps the translator translate the Source text correctly, for example when the Source text contains "Data placeholders in translation entries" on page 471 or "HTML tags in translation entries" on page 472.

Keyboard shortcuts

This topic gives an overview of keyboard shortcuts that can be used in the Designer.

Although some of the keyboard shortcuts are the same, this isn't a complete list of Windows keyboard shortcuts. To find that, refer to the documentation of the Windows version that you are using.

Menu items

The following key combinations activate a function in the menu.

Key combination	Function
Alt	Put the focus on the menu. (Alt + the underlined letter in a menu name displays the corresponding menu.) The menu can then be browsed using the Enter key, arrow up and

Key combination	Function
	arrow down buttons.
Alt + F4	Exit
Alt + Shift + D	"Design tab" on page 595
Alt + Shift + E	Preview HTML
Alt + Shift +L	Live view
Alt + Shift + P	"Preview tab" on page 596
Alt + Shift + S	"Source tab" on page 597
Alt + Shift + T	Tag element (see "Tagging elements for translation" on page 469)
Ctrl + C or: Ctrl + Insert	Copy
Ctrl + F	Find
Ctrl + I	Properties
Ctrl + N	New
Ctrl + O	Open file
Ctrl + P	Print
Ctrl + R	Show/hide rulers
Ctrl + S	Save file
Ctrl + V	Paste

Key combination	Function
or: Shift + Insert	
Ctrl + X or: Shift + Delete	Cut
Ctrl + W	Close file
Ctrl + Y or: Ctrl + Shift + Z	Redo
Ctrl + Z	Undo
Ctrl + Alt + ;	Lock guides
Ctrl + Shif + R	Clear preview cache
Ctrl + Shift + S	Save all
Ctrl + Shift + W	Close all
Ctrl + Shift + ;	Snap to guides
Ctrl + ;	Show guides
Ctrl + '	Show/hide virtual stationery
Ctrl + \	Highlight Master Page items
Ctrl + F5	Revert
Ctrl + F10	Save as
Ctrl + Shift + F12	Package

Workspace

The following key combinations activate a function in the Workspace.

Key combination	Function
Alt + -	Open system menu
Alt + F7 or: Alt + Page Down	Next tab
Alt + Shift + F7 or: Alt + Page Up	Previous tab
Ctrl + A	Select all
Ctrl + Shift + E	Switch to Editor
Ctrl + F	Find (opens the Find and Replace dialog: "Find/Replace Dialog" on page 497)
Ctrl + F6	Next editor (when there is more than one file open in the Workspace)
Ctrl + Shift + F6	Previous editor (when there is more than one file open in the Workspace)
Shift + F10 or: Ctrl + Shift + F10	Open context menu

Design and Preview tab

In addition to the keyboard shortcuts for menu items and the Workspace, the following key combinations have a special function in the **Design** tab and **Preview** tab of the Workspace.

Key combination	Function
Ctrl + B	Bold (works on a text selection)
Ctrl + E	Open Paragraph formatting dialog
Ctrl + I	Italic (works on a text selection)
Ctrl + H	Show Edges
Ctrl + K	Delete browser element
Ctrl + M	Open Box formatting dialog
Ctrl + T	Open Text formatting dialog
Ctrl + U	Underline ((works on a text selection)
Ctrl + **+** or: Ctrl + Shift + **+** or: Ctrl + = or: Ctrl + Shift + =	Zoom in **Tip** You can also use the mouse's scroll wheel in combination with the Ctrl button to gradually zoom in or out.
Ctrl + **-** or: Ctrl + Shift + **-**	Zoom out
Ctrl + 0	Zoom to page width
Ctrl + 1	Zoom to page content width
Ctrl + arrow up	Select parent
Ctrl + Shift + R	Refresh

Key combination	Function
or: F5	

Text editors: Source tab, JavaScript, CSS, Script Editor

The following key combinations have a special function in the **Source** tab of the Workspace (see also: "Source tab" on page 597), and when editing a JavaScript or CSS file in the Workspace, and in the Script Editor (expanded view).

Key combination	Function
Ctrl + space	Content assist (auto-complete)
Ctrl + A	Select all
Ctrl + D	Duplicate line
Ctrl + F	Find
Ctrl + I	Indent (Tab)
Ctrl + J	• Script Editor: Add a line break. • Other editors: Incremental find; start typing a search string directly after pressing this key combination.
Ctrl + K	Find next
Ctrl + L	Go to line; a prompt opens to enter a line number.
Ctrl + Shift + D	Delete line
Tab	Expand Emmet abbreviation (see "Emmet preferences" on page 382)
Shift + Tab	Shift selected lines left

Key combination	Function
Tab	Shift selected lines right
Ctrl + /	Comment out / uncomment a line in code
Ctrl + Shift + /	Comment out / uncomment a code block

Scripts pane and Resources pane

The following keys or key combinations have a special function when a file is selected in the **Resources** pane and when a script is selected in the **Scripts** pane.

Key combination	Function
F2	Rename
Alt + Enter	Open Properties dialog
Delete	Delete

Data Model pane

You can use the following keys to browse records in the Data Model pane:

- Page Up: next record
- Page Down: previous record
- Home: first record
- End: last record.

Menus

The following menu items are shown in the Designer menu. For a list of keyboard shortcuts, see "Keyboard shortcuts" on page 561.

File Menu

- **New...**: Opens the **New (Select a Wizard)** dialog that lets you create a template (see "Templates" on page 109).
- **Open**: Opens a standard File Open dialog. This dialog can be used to open templates. See "Templates" on page 109. .
- **Open Recent**: Lists the most recently opened templates.
- **Close**: Closes the currently active template. If the file needs to be saved, the appropriate Save dialog will open.
- **Close All**: Closes any template. If any of the files needs to be saved, the Save Resources dialog opens.
- **Close Others**: Closes all files except the one that is currently active in the workspace.
- **Save**: Saves the current file to its current location on disk. If the file has never been saved, the Save As dialog appears instead.
- **Save All**: Saves the open files. If a file has never been saved, the Save As dialog opens for it.
- **Save As...**: Saves the current file to a new location on disk.
- **Save a Copy**: Save a copy of the current template in the selected Connect version's format. See "Saving a copy / down-saving a template" on page 112.
- **Revert**: Reverts all changes to the state in which the file was opened or created.
- **Add Data**: Adds data to the open template. See "Loading data" on page 341 .
  - **From File Data Source...**: Opens the dialog to add a new data file. .
  - **From Database Data Source...**: Opens the Edit Database Configuration dialog.
  - **Generate Counters**: Opens the Generate Counter Wizard to create a custom counter as a data source.
  - **JSON sample data...**: Opens the dialog to load JSON sample data in the Data Model pane; see "JSON sample data dialog" on page 505.
- **Package...**: Opens the dialog that lets you package files; see "Package dialog" on page 520.
- **Print**: Opens the "Print options" on page 704 dialog.
- **Print Presets**: Selecting this option allows you to create or modify Print Presets, which can be saved and used in print runs thereafter.
  - **Job Creation Presets**: Opens the"Job Creation Presets Wizard" on page 684 dialog.

    - **Output Creation Presets**: Opens the "Output Creation Presets Wizard" on page 700 dialog.
- **Proof Print**: Opens the "Print options" on page 704 dialog as a Proof Print dialog which limits the number of records output. The options themselves are identical to the regular Print Output dialog.
- **Send Email**: Opens the Send Email dialog; see "Send (Test) Email" on page 543 and "Generating Email output" on page 1017.
- **Send Test Email**:Opens the Send Test Email dialog; see "Send (Test) Email" on page 543.
- **Exit**: Closes the software. If any of the files needs to be saved, the Save Resources dialog opens.

Edit Menu

- **Undo <action>**: Undoes the previous action that was done.
- **Redo <action>**: Redoes an action that was previously undone.
- **Cut**: Cuts the currently selected text, object or element and puts it on the clipboard.
- **Copy**: Copies the currently selected text, object or element to the clipboard.
- **Copy to snippet**: Creates a new snippet from the selected text, object or element.
- **Paste**: Takes the current clipboard content and pastes it at the pointer location.
- **Paste as Plain Text**: Takes the current clipboard content and pastes it at the pointer location without any HTML styles or formatting.
- **Delete Browser Element**: Removes the currently selected element in the workspace.
- **Find/Replace**: Only active while inside the Workspace. Opens the Find/Replace dialog.
- **Stylesheets...**: Open the "Style sheets dialog" on page 548. See "Styling and formatting" on page 294 and "Styling templates with CSS files" on page 296.
- **Colors...**: Opens the Colors Editor dialog. See "Colors" on page 328.
- **Fonts...**: opens the "Font Manager" on page 499. See "Fonts" on page 332.
- **Locale...**: Opens the Locale Settings dialog. See "Locale" on page 336.
- **Color Settings...**: Opens the Color Settings dialog. See "Colors" on page 328.

Insert Menu

- **Image**: Inserts an image using a resource that is local to the template, a resource on disk or a URL. See "Images" on page 271.

- **Text**:
  - **Wrap in span**: Wraps selected text in a element. The ID or class of the span can be used as a selector for scripts and styles.
- **Special Characters**: Displays a categorized list of special HTML characters that can be inserted at the current pointer location. When a character is clicked, its HTML Entity is inserted. This includes:
  - **Symbols**: Use the list to insert a special symbol such as Copyright, Trademark, or Ellipsis.
  - **Markers**: Use the list to insert pagination markers that are replaced with specific page numbering:
    - **Page Number**: This marker is replaced by the current page number in the document. Even if the page number is not used on certain pages, those page are still added to the page count.
    - **Page Count**: This marker is replaced by the total number of pages in the document, including pages with no contents.
    - **Content Page Number**: This marker is replaced by the current page number (with contents) in the document.
    - **Content Page Count**: This marker is replaced by the total number of pages that have contents in them, in the document. A page with contents is a page that is part of a section that has variable data on it. A page with a Master Page but no contents (set in the Sheet Configuration tab of the "Print section properties" on page 542) is not included in the Content Page Count.
    - **Sheet Number**: This marker is replaced by the current sheet number (physical piece of paper with two sides, or pages) in the document. This is equivalent to half the page number, for example if there are 10 pages, there will be 5 sheets.
    - **Sheet Count**: This marker is replaced by the total number of sheets in the document, whether or not they have contents.
  - **Dashes and Spaces**: Use the list to insert special dashes, such as an em-dash, and spaces, such as non-breaking spaces or an en-space. (The HTML code inserted for the dash or space is visible on the Source tab of the workspace.)
  - **Arrows**: Use the list to insert directional arrows (in one of four directions).
  - **Geometric Shapes**: Use the list to insert a special geometric shape, such as circles, triangles and squares.
- **Date**: Opens the "Date" on page 267 dialog to add a date to the template based on the current system's date and time.

- **Wrap in box**: Puts the element in which the cursor is located in an inline box (a).
- **Table**
  - **Thead, tbody, tfoot**: Insert a header, body or footer (if not already present) in the current table.
  - **Standard**: Inserts a table with a specific number of columns and rows through the Standard Table Wizard; see "Table" on page 278.
- **Table Elements**:
  - **Insert Row Above**: Inserts a row above the current one. The row configuration, such as merged cells and cell styles, is duplicated, but contents is not.
  - **Insert Row Below**: Inserts a row below the current one. The row configuration, such as merged cells and cell styles, is duplicated, but contents is not.
  - **Insert Column Before**: Inserts a column to the left of the current one. The column configuration, such as merged cells and cell styles, is duplicated, but contents is not.
  - **Insert Column After**: Inserts a column to the right of the current one. The column configuration, such as merged cells and cell styles, is duplicated, but contents is not.
- **Common Elements**:
  - **Paragraph...**: Opens a dialog to add a <p> element; see "Text and special characters" on page 283.
  - **H1 through H6...**: Opens a dialog to add a <h1> to <h6> element; see "Text and special characters" on page 283.
  - **Address...**: Opens a dialog to add an <address> element.
  - **Preformatted...**: Opens a dialog to add a <pre> element.
- **Structural Elements**:
  - **Div**...: Opens a dialog to add a element; see "Boxes" on page 256
  - **Span**...: Opens a dialog to add a element; see "Boxes" on page 256
  - **Article**...: Opens a dialog to add an <article> element
  - **Section**...: Opens a dialog to add a <section> element (the HTML element, not a section in a context).
  - **Header**...: Opens a dialog to add a <header> element.
  - **Footer**...: Opens a dialog to add a <footer> element.
  - **Nav**...: Opens a dialog to add a <nav> element.

- **Aside**...: Opens a dialog to add an <aside> element.

> **Note**
>
> Article, Section, Header, Footer, Nav and Aside are HTML5 semantic elements; see https://www.w3schools.com/html/html5_semantic_elements.asp

- **Business Graphic**: Displays a list of available business graphic object to be inserted:
  - **Insert Pie Chart**: Opens the Pie Chart script dialog to insert a new Pie Chart.
  - **Insert Bar Chart**: Opens the Bar Chart script dialog to insert a new Bar Chart.
  - **Insert Line Chart**: Opens the Line Chart script dialog to insert a new Line Chart.
- **Barcode**: Displays a list of available barcodes. Click on one to insert it in the page. See "Barcode" on page 197.

Format Menu

- **Size**: When text is selected, choose a predefined or custom font size in this submenu to change the size of the selected text.
  - **Other...**: Opens the Text Formatting dialog for advanced style selection; see "Styling text and paragraphs" on page 308.
  - **7pt - 72pt**: Sets the size of the selected text to the chosen font size.
- **Style**: When text is selected, sets the text style by applying or removing the following attributes: Plain, Bold, Italic, Underline, Strikethrough, Subscript, Superscript, Capitalize, Uppercase, Lowercase, Small-caps. This is the same as opening the Text Formatting dialog (**Format > Text**) and checking the appropriate style. See "Styling text and paragraphs" on page 308.
- **Color**: When text is selected, sets the text color by applying the color attribute to the text. The color submenu lists all the colors in the Colors Editor.
- **Text...**: Opens the Text Formatting dialog to modify the current text selection. See "Styling text and paragraphs" on page 308.
- **Align**: When an element is selected, determines how its contents is aligned inside the element. Options are Align Left, Align Right, Align Center and Justify.
- **Paragraph...**: Opens the "Paragraph Formatting dialog" on page 520 to modify the paragraph where the cursor is located. See "Styling text and paragraphs" on page 308.

- **Paragraph Format**: Displays a list of generic element types that can be used for a text element. Selecting one of them converts the element where the cursor is located into the appropriate element (for example <p> for Paragraph, <h3> for Heading 3, etc).
- **Float**
  - **Left**: Floats the current element to the left. This is equivalent to setting the CSS `float` property to `left`.
  - **Right**: Floats the current element to the right. This is equivalent to setting the CSS `float` property to `right`.
  - **None**: Removes any float style applied to the currently selected element.
- **Align Objects**:
  - **Top**: Aligns the top side of the selected objects with the top edge of the last selected object.
  - **Middle**: Aligns the middle of the selected objects with the middle of the last selected object. Objects may move up or down.
  - **Bottom**: Aligns the bottom side of the selected objects with the bottom edge of the last selected object.
  - **Left**: Aligns the left side of the selected objects with the left edge of the last selected object.
  - **Center**: Aligns the center of the selected objects with the vertical center of the last selected object. Objects may move to the left or to the right.
  - **Right**: Aligns the right side of the selected objects with the right edge of the last selected object.
- **Box...**: Opens the "Box Formatting dialog" on page 487 to modify the box where the cursor is located.
- **Image...**: Opens the "Image Formatting dialog" on page 500 to modify the image that is currently selected.
- **Table...**: Opens the "Table Formatting dialog" on page 553 to modify the table in which the cursor is located. If the cursor is within a table embedded within another, the innermost table's formatting is the one modified.
- **Table Cell...**: Opens "Table Cell Formatting dialog" on page 557 to modify the cell where the cursor is located.
- **Hyperlink**
  - **Insert...**: Creates a hyperlink on the currently selected text or element and opens its properties; see "Hyperlink and mailto link" on page 268.

- **Edit...**: Opens the properties for the currently selected hyperlink; see "Hyperlink and mailto link" on page 268.
- **Remove**: Removes the currently selected hyperlink. The text or element that was the hyperlink is not removed.

Context Menu

- **Add**:
  - **Print Context**: Adds a new Print context to the template if one does not exist.
  - **HTML Email Context**: Adds a new Email context to the template if one does not exist.
- **Delete**: Deletes the currently selected context. The last remaining context cannot be deleted.
- **Go to**: Opens the first section in the selected context. This is the same as double-clicking on the first section of any context in the Resource Pane.
- **Finishing**: Opens the Print context's Finishing options dialog (see "Setting the binding style for the Print context" on page 135). This option is only available when editing a Print section in the Workspace.
- **PDF Attachments**: Opens a dialog to set the compression for PDF attachments; see "PDF Attachments dialog" on page 523. This option is only available when editing an Email section in the Workspace.
- **Preview HTML**: Opens the currently selected section in the default system browser to preview it. This feature works in all contexts.
- **Profile Scripts**: Opens the "Profile Scripts dialog" on page 530 to test script performance (see "Testing scripts" on page 403).
- **Preflight**: Opens the Preflight dialog. Preflight checks the template for common errors (see "Testing scripts" on page 403).

Section Menu

- **Add**: Adds a new section to the currently selected context.
- **Delete**: Deletes the currently selected section.
- **Arrange**: Opens the "Arrange Sections" on page 543 dialog.
- **Go to**: Lists the sections in the currently selected context. Open one by clicking it.
- **Properties...**: Opens the appropriate section properties: Email or Print. See "Section properties dialogs" on page 540.

- **Includes...**: Opens the "Includes dialog" on page 504.
- **Finishing...** (Print Sections only): Opens the Finishing tab in the "Print section properties" on page 542.
- **Sheet Configuration...** (Print Sections Only): Opens the "Sheet Configuration dialog" on page 546.
- **Master Pages**: L ists the available Master Pages in the template (see "Master Pages" on page 156). Open one by clicking it.
- **Master Page Properties**...: Opens the currently selected Master Page's properties dialog; see "Master Pages" on page 156.

View Menu

- **50/75/100/150/200%**: Zooms the Workspace at the selected level.
- **Source View**: Shows the HTML source for the template, including CSS and HTML code.
- **Design View**: Shows the template including all styles, text and images as well as the placeholders used for variable data.
- **Preview View**: Shows the template as it will output with the current record, with the personalized content (see "Personalizing content" on page 338).
- **Refresh**: Reloads the view, including static external images and remote stylesheets, and re-runs the scripts (the latter in Preview Mode only).
- **Show Edges**: Shows or hides a colored border around elements on the page.
- **Rulers**: Shows or hides the rulers in the Workspace. Rulers only appear for Print contexts.
- **Guides**:
  - **Show Guides**: Shows or hides the margin lines and guides in a Print section (see "Print" on page 124, "Page settings: size, margins and bleed" on page 148 and "Guides" on page 314). The colors of margin lines and guides are adjustable; see "Editing preferences" on page 377.
  - **Lock Guides**: Locks the guides, so that they cannot accidentally be moved while working on the Print context.
  - **Snap to Guides**: Enables or disables snapping to guides and to margins when moving objects.
- **Virtual Stationery**: Enables or disables the visibility of the PDF Background image set in the Media.

- **Highlight Master Page Items**: Enables or disables a yellow border around Master Page items in a section.
- **Object Resizing**: Enables or disables the ability to resize elements on the page. See "Editing preferences" on page 377 for more fine-tuned control.
- **Shared Content Editing**: Enables or disables the ability to edit shared content in locations where it is used. See "Snippets" on page 285.

Window Menu

- **Show View**>: Use the options in this menu to show or hide different panes of the UI.
  - **Translations**: Shows the "Translations pane" on page 594.
  - **Outline**: Shows the "Outline pane" on page 583.
  - **Messages**: Shows the Messages pane, see "Preflight Results and Messages" on page 581.
  - **Data Model**: Shows the "Data Model pane" on page 579.
  - **Preflight Results**: Shows the results of a Preflight; see "Doing a Preflight" on page 405.
  - **Scripts**: Shows the "Scripts pane" on page 590.
  - **Parameters**: Shows the Parameters pane. See Runtime parameters.
  - **Styles**: Shows the "Styles pane" on page 593.
  - **Attributes**: Shows the "Attributes pane" on the next page.
  - **Resources**: Shows the "Resources pane" on page 583.
- **Reset Perspective**: Resets all toolbars and panes to the initial configuration of the module.
- **Clear Recent Files Lists**: Clears the lists of the most recently opened templates and data mapping configurations under File > Open Recent and in the Welcome screen.
- **Preferences**: Click to open the Preferences dialog.

Help Menu

- **Software Activation**: Displays the Software Activation dialog. See "Activating a License" on page 45.
- **Help Topics**: Opens the help system in the default web browser.
- **Contact Support**: Opens the Objectif Lune Contact Page in the default system web browser.

- **About PrintShop Mail Connect Designer**: Displays the software's About dialog.
- **Welcome Screen**: Re-opens the Welcome Screen.

Panes

Panes are windows containing user interface elements (such as information or properties), which can be docked and undocked, moved around and merged together through tabbed panes.

Here is a list of all panes:

Attributes pane

The Attributes pane displays all of the properties of the currently selected object in the Workspace. These properties vary greatly depending on the object that has been selected.

General

These attributes are common to all elements in the template and will always appear.

- **ID**: A unique identifier for the selected element. Used for CSS selections (see "Styling templates with CSS files" on page 296) as well as scripts (see "Writing your own scripts" on page 393) affecting **single** elements.
- **Class**: One or more classes that can be common to more than one elements. Used for CSS selections and scripts that can affect **multiple** elements.

> **Note**
>
> Do not give an element the ID 'pages' or the class name 'dynamic'. These are reserved words. Using them as an ID or class name leads to undesirable effects.

Other

These attributes are available depending on the item selected.

- **Allow Resizing**: Allows columns in a **table** to be resized directly within the Workspace in Design mode.

- **Alternate text**: The "Alt" text used when hovering over the **image** in a browser. Also used for accessibility.
- **Cellspacing**: Defines the *cellspacing* attribute of the **table** which controls the spacing between cells in the table.
- **Cellpadding**: Defines the *cellpadding* attribute of the **table** which controls the padding inside each cell of the table.
- **Source**: The location of the **image** file. For image resources in the template, the image path is often `images/<imagefile>.<extension>`
  When the source is a PDF, an addition button appears next to this box that opens the "Select Image dialog" on page 545.
- **Sum**: Adds a running total (i.e. a subtotal) to a cell in a Dynamic Table. See Adding subtotals and transport lines.
- **Type (form input element)**: Use the drop-down to select an input type. The drop-down lists all input types, including HTML5 input types (see https://www.w3schools.com/html/html_form_input_types.asp).
- **Whitespace element**: Check to make the element a whitespace element, meaning it will only appear on the page if there is enough space for it. This is useful for templates with variable height elements or conditional elements, to fill empty spaces with transpromotional material. Note that only top-level elements (i.e. not inside another element such as a table or div) will function as whitespace elements.

Geometry

These attributes are available for certain elements that have position or size attributes such as images and boxes.

- **Top**: The horizontal distance from the top-left of the object to the left position of its parent. This is used only for relative and absolute positioned elements.
- **Left**: The vertical distance from the top-left of the object to the top position of its parent.
- **Width**: The width of the element, by default in pixels. For an image, this defaults to the original image width in pixels.
- **Height**: The height of the element, by default in pixels. For an image, this defaults to the original image height in pixels.
- The **Reset Image Size** button resets the selected image to its original size.

Note

When no unit is added to a geometry value, the default unit will be added to the value; see "Print preferences" on page 390.

Page

These attributes appear when selecting the *Page* node in the Outline Pane.

- **Master Page**: Which of the "Master Pages" on page 156 to use for the template.

Data Model pane

The Data Model Pane displays a Data Model used to help design the template, along with data. When you load data (see "Loading data" on page 341), the resulting record set is loaded in the Data Model pane. The information shown is the information for the current record within the record set.

Filter

To find a certain table, group or field in a large Data Model, start typing characters in the Filter box. This immediately narrows down the list of displayed items to all tables, groups and fields whose **name** contains the characters that were typed. Note that the filtering is case-insensitive.

Data Model toolbar buttons

- **JSON Sample Data**: Opens a dialog to load JSON sample data in the Data Model pane; see "Adding JSON sample data" on page 349.

  Tip

  If a data field contains JSON data you could use that data as sample data: right-click the field, select **Copy** to copy the JSON data to the clipboard; then open the JSON Sample Data dialog and paste the data there.

- **Minimize/Maximize**: Click to minimize or maximize the pane. See Moving and Merging Panes.

Browsing data

When a Data Model is loaded in the Data Model Pane, it can be used to design templates by dragging fields directly into the template; see "Variable Data" on page 351.

If data is present it is possible to preview the resulting data in the template on the **Preview** tab (see Workspace).

You can use the following keys to browse records in the Data Model pane:

- Page Up: next record
- Page Down: previous record
- Home: first record
- End: last record.

The way the Data Model is displayed can be changed via the following options in the contextual menu. Right-click the Data Model to open the contextual menu.

- **Collapse All**: Collapse all fields.
- **Expand All**: Expand all fields.

The record can also be expanded and collapsed using the arrow icon next to it.

> **Tip**
>
> Right-click a field and select **Copy** to copy its contents to the clipboard.

Preflight Results and Messages

Messages pane

The **Messages** pane displays any warnings and errors from the data mapping configuration or template.

To open it in the Designer module, click the Messages button at the bottom right of the window (see "Designer User Interface" on page 476).

Buttons

- **Export Log** : Click to open a **Save As** dialog where the log file (.log) can be saved on disk.
- **Clear Log Viewer** : Click to remove all entries in the log viewer.
- **Filters** : Displays the Log Filter.
- **Activate on new events** : Click to disable or enable the automatic display of this dialog when a new event is added to the pane.

Field headers

- **Message**: The contents of the message, indicating the actual error.
- **Component**: Whether the entry is a warning or an error.
- **Source**: The source of the error.
- **Date**: The date and time when the error occurred.

Preflight Results pane

The Preflight Results pane displays any notifications or errors related to the template, its scripts, its code or output generation. Double-click a script warning/error to open the script in the script editor. The relevant line will be highlighted.
See also: "Testing scripts" on page 403.

Moving and merging panes

The PrintShop Mail Connect interface for the Designer module is highly configurable. Each panel in the application can be moved, with the exception of the "Workspace" on page 595 which area always in a static location. All panels can be minimized or maximized.

To move a panel:

- Click and hold the left mouse button on the panel title (tab) to move and keep the button pressed.
- Start moving the mouse to the new location. A grey outline shows where the tab will show up:
  - A small grey outline next to a current panel tab indicates that both tabs will be at the same location and only the active tab will display its content.

    - At larger grey outline at one of the edges of the Workspace or Data Viewer indicates that the separate will be separate and always visible.
- When the grey outline displays the location where the panel should be, release the mouse button.

To minimize a panel:

- Click the **Minimize panel** button at the top-right corner of the panel.

A minimized panel displays only as its icon wherever it was docked, generally on the left or right side, or the bottom.

To restore a minimized or maximized panel:

- Click the **Restore** button next to the panel's display icon.

The restored panel will return to its original docked location.

To temporarily display a minimized panel:

- Click the panel's display icon.

When another panel, menu or toolbar is clicked, the panel will be minimized again.

To maximize a panel:

- Click the **Maximize** button at the top-right corner of the panel.

A maximized panel takes the full available size for the panels. All other panels are minimized.

Outline pane

The Outline pane displays the current structure of the template, including all HTML tags present in each page.

- The display is in a treeview, the root being the *Page* node.
- At the top of the pane, a Text Filter box appears. Enter text in this box to only show elements which correspond to this inclusive filter. This can be class names, IDs, or element types (div, table, etc).

- Under the **Page** node, all top-level page elements are displayed. Each element under them is accessible by expanding (with the [+]) elements with children.
- Clicking on any element will select it in the Workspace, whether it displays the Source, Design or Preview tab.
- Dragging an element inside the Outline pane re-orders it in the actual HTML. Elements are executed top-to-bottom with lower elements appearing on top of previous elements (unless a CSS Z-Index is used).
- Right-clicking an element displays a contextual menu offering the following option:
  - **Delete Element**: Click to delete the element from the outline view. This also removes it in the template itself for the current section.

Resources pane

The Resources pane displays the resources that affect the template and its output.

> **Tip**
>
> The Text Filter box at the top of the Resources pane allows to look for text in the name of resources and in the source of any text-based files: HTML (including sections, master pages, and snippets), JSON, JS and CSS.

> **Tip**
>
> Images, fonts, stylesheets and snippets can be dragged or copied and pasted into the Resources pane to add them to your template.

Media

Media resources define paper handling configurations for Print output (see "Generating Print output" on page 993 and "Print options" on page 704) including page size and paper type. See "Media" on page 159 for more information.

Contextual menu

- **New Media**: Click to create a new Media and open its properties ("Media Properties" on page 516).

- **Delete**: Click to delete the resource. This is the same as pressing the Delete key while the resource is selected.
- **Rename**: Click to open the resource's Rename. This is the same as pressing the F2 key while the resource is selected.
- **Properties**: Click to open the media properties.
- **Copy**: Click to copy the selected Media to the clipboard.
- **Paste**: Paste the copy into the Media folder.

Master Pages

Master Pages are layers of content that can be used by multiple Print Contexts to provide a reusable static background of content. Only one Master Page can be selected for each page position in the context. See "Master Pages" on page 156 for more information.

Contextual menu

- **New Master Page**: Click to create a new Master Page and open its properties.
- **Rename**: Click to open the resource's Rename. This is the same as pressing the F2 key while the resource is selected.
- **Delete**: Click to delete the resource. This is the same as pressing the Delete key while the resource is selected.
- **Properties**: Click to open the Master Page properties; see "Master Pages" on page 156 for more information.
- **Copy**: Click to copy the selected Master Page to the clipboard.
- **Paste**: Paste the copy into the Master Pages folder.

Contexts

Contexts hold the actual content of the template that is used to generate output. See "Contexts" on page 118 for more information.

Contextual menu (Context folder or individual contexts)

- **New Print Context**: Click to create a new Print Context with a single section.
- **New HTML Email Context**: Click to create a new HTML Email context with a single section.

- **Properties...** (Print and Email Contexts): Click to open the Context's properties. See "Contexts" on page 118 for more information.
- **Paste**: Paste the copied section into the Context folder.

Sections

Sections hold part of the contents within a specific context. See "Sections" on page 119 for more information.

Contextual menu

- **Set as Default** (Email contexts only): Click to set the default section that is output if none is selected in the output generation.
- **New Section**: Click to add a new section within the context.
- **Rename**: Click to open the resource's Rename. This is the same as pressing the F2 key while the resource is selected.
- **Delete**: Click to delete the resource. This is the same as pressing the Delete key while the resource is selected.
- **Properties...**: Click to open the appropriate section properties: Email or Print. See "Section properties dialogs" on page 540.
- **Includes...**: Click to open the "Includes dialog" on page 504.
- **Finishing...** (Print Sections only): Click to open the Finishing tab in the "Print section properties" on page 542
- **Sheet Configuration...** (Print Sections Only): Click to open the Sheet Configuration dialog; see "Master Pages" on page 156 and "Media" on page 159.
- **Copy**: Click to copy the selected section (including its settings) to the clipboard.
- **Paste**: Paste the copied section into the same Context.

Images

Images are graphical elements that can be added to the page for display, either statically or dynamically. See "Images" on page 271 for more information.

Contextual menu

- **New Folder**: Click to create a new folder to organize resources more easily.
- **Rename**: Click to open the resource's Rename dialog. This is the same as pressing the F2 key while the resource is selected.
- **Delete**: Click to delete the resource. This is the same as pressing the Delete key while the resource is selected.
- **Copy**: Click to copy the selected file.
- **Copy Resource Location**: Click to copy the relative path of the resource to the clipboard. It may then be pasted into a script.

Fonts

Font Resources included in a template are transported with it, so they can be accessed even if the template is moved to a different computer. Fonts appear in the fonts drop-down menu but they can also be set through a CSS Stylesheet.

Currently supported font types: otf, woff, ttf, svg. Fonts must be set to *installable* to be useable in the output.

See "Fonts" on page 332.

JavaScripts

See Using JavaScript for more information.

JavaScript files that are linked to (i.e. included in) a section show a chain icon.

Contextual menu

- **New Javascript**: Click to create a new JavaScript resource.
- **New Remote Javascript**: Click to add a Remote JavaScript resource. See Using JavaScript for more information.
- **New Folder**: Click to create a new folder to organize resources more easily.
- **Rename**: Click to open the resource's Rename. This is the same as pressing the F2 key while the resource is selected.

- **Download Resource** (remote JavaScript files only): Select to download the Remote JavaScript file (see Adding a remote JavaScript file) and add the downloaded copy to the Scripts folder. The copy will have the same name as the remote script file, but its extension will be .js instead of .rjs. Any section that used the JavaScript file will from then on use the downloaded file (see Including a JavaScript file in a Web context).
- **Delete**: Click to delete the resource. This is the same as pressing the Delete key while the resource is selected.

Stylesheets

Style sheets control how contents appears on the page. They define spacing, color, size and other properties of elements on the page. See "Styling templates with CSS files" on page 296 for more information.

In case the CSS file has references to specific images, you can drag/drop or copy/paste those images into the Stylesheets folder as well.

Style sheets that are linked to (i.e. included in) a section show a chain icon. See "Applying a style sheet to a section" on page 304.

Contextual menu

- **New Stylesheet**: Click to create a new Stylesheet resource. Adding a new stylesheet will automatically include it in the currently active section.
- **New Remote Stylesheet**: Click to add a Remote Stylesheet resource. See "Styling templates with CSS files" on page 296 for more information.
- **New SCSS file**: Click to add a new, empty Sass file to the Resources (extension: .scss). **Sass** is a CSS Preprocessor integrated in Connect. For more information about Sass, see: Sass website.

> **Note**
>
> When the name of Sass file begins with an underscore, it is considered a partial .scss file (e.g. _mySass.scss). Partial files are typically imported in a base .scss file. They may include Sass variables or other directives declared in the base file, and they cannot be compiled.

- **New Remote SCSS file**: Click to add a Remote Sass file to the Resources.
- **Compile** (.scss files only): Select to compile the Sass (.scss) file into a Stylesheet which can then be applied to a section.
  The compiled style sheet will have the same name as the Sass file, but with the extension .css.
  Compiled CSS files can be recognized by their first line: /* Compiled by https://sass-lang.com/libsass */.
  Compiler options can be set in the Preferences; see "Editing preferences" on page 377.

  > **Note**
  >
  > Re-compiling a .scss file overwrites any manual changes made to the .css file.

  > **Note**
  >
  > Partial .scss files cannot be compiled.

- **New Folder**: Click to create a new folder to organize resources more easily.
- **Rename**: Click to open the resource's Rename. This is the same as pressing the F2 key while the resource is selected.
- **Download Resource** (remote style sheets only): Select to download the Remote Stylesheet (see "Using a remote style sheet" on page 299) and add the downloaded copy to the Stylesheets folder. The copy will have the same name as the remote style sheet, but its extension will be .css instead of .rcss. Any section that used the remote style sheet will from then on use the downloaded style sheet (see "Applying a style sheet to a section" on page 304).
- **Delete**: Click to delete the resource. This is the same as pressing the Delete key while the resource is selected.

Snippets

A snippet is a small, ready-to-use piece of content in a file. Snippets can be re-used within the same template, in all contexts, sections and scripts.

Contextual menu

- **New HTML Snippet**: Click to create a new HTML Snippet resource; see "Snippets" on page 285.
- **New JSON Snippet**: Click to create a new JSON Snippet resource; see "JSON snippets" on page 290.
- **New Handlebars template**: Click to create a new Handlebars template; see "Handlebars templates" on page 291.
- **New Folder**: Click to create a new folder to organize resources more easily.
- **Rename**: Click to open the resource's Rename. This is the same as pressing the F2 key while the resource is selected.
- **Delete**: Click to delete the resource. This is the same as pressing the Delete key while the resource is selected.
- **Copy**: Click to copy the selected file.
- **Copy Resource Location**: Click to copy the relative path of the resource to the clipboard. It may then be pasted into a script.

Translations

A **PO file** holds translated texts for the template in one language. Each PO file has an entry by which OL Connect recognizes the target language.
PO files can be stored in the Translations folder.

See "Translating templates" on page 467 and "Importing translations (.po)" on page 475 for more information.

Contextual menu

- **New Translation**:
  - **PO files...**: Select a PO file to import it into the template.
  - **Remote PO file**: Enter the URL Click to add a remote PO file. See Using JavaScript for more information.
- **New Folder**: Click to create a new folder to organize resources more easily.
- **Rename**: Click to open the resource's Rename. This is the same as pressing the F2 key while the resource is selected.

- **Download Resource** (remote JavaScript files only): Select to download the Remote JavaScript file (see Adding a remote JavaScript file) and add the downloaded copy to the Scripts folder. The copy will have the same name as the remote script file, but its extension will be .js instead of .rjs. Any section that used the JavaScript file will from then on use the downloaded file (see Including a JavaScript file in a Web context).
- **Delete**: Click to delete the resource. This is the same as pressing the Delete key while the resource is selected.

Scripts pane

The Scripts pane contains all of the scripts that are used to replace data in a template, or to modify its look; see "Personalizing content" on page 338.
The scripts are stored in three groups: Control, Standard and Post Pagination, depending on when they run (see "The script flow: when scripts run" on page 413).

> **Note**
>
> The scripts in each of the main folders - Control, Standard and Post Pagination - are executed top-to-bottom. They can be dragged up or down in the pane to change their order of execution. For example, content-loading scripts (snippets with variable data, for instance) must come before scripts that replace data within loaded contents.

The Script pane allows to add and edit scripts and to manage scripts, or instance by organizing them in subfolders and enabling/disabling them (see "Managing scripts" on page 400).

Scripts can be exported and imported via the buttons or through drag & drop between the Scripts pane and any location on the computer.

> **Note**
>
> Scripts included on the Scripts pane are completely distinct from the JavaScript resources found in the "Resources pane" on page 583 (see Using JavaScript).
> Think of scripts as server-side in the sense that they are executed through the Connect modules (Server and Content Creation especially). Scripts have access to the whole PrintShop Mail Connect JavaScript API (see "Standard Script API" on page 810), such as the
> `record`

object.
JavaScript resources, on the other hand, are only executed **after** the content creation is done, generally in a browser.

Buttons

- **Import...**: Click to open a standard Open dialog to import a script. The script must have the .OL-script extension.
- **Export...**: Click to open a standard Save As dialog to save the currently selected scripts to disk. These scripts can be re-used in other templates. If more than one script is selected, they are all saved to a single file. If some scripts are inside folders, this folder structure is kept and will be restored when the scripts are imported.
- **New**: Displays a drop-down that shows the following options:
  - **Control Script**: A Control Script affects the output of a template per record as a whole, instead of parts of the content. See "Control Scripts" on page 429 and "Control Script API" on page 931.
  - **Standard Script**: Adds a new empty standard script. Double-clicking the new script opens the script editor, where you can write code (see "Writing your own scripts" on page 393).

    Tip

    Select **Text Script** instead of Standard Script to create a Standard script that is opened with the Script Wizard.

  - **Post Pagination Script** (Print context only): Adds a Post Pagination script, which runs on the Print context after the output has been paginated (see "Post Pagination Scripts" on page 445).
  - **Folder**: Adds a folder in which scripts can be placed for easier management. See "Script folders" on page 400.
  - **Text Script**: When you add variable data to a template, a script of this type is created by default. See "Variable Data" on page 351.
  - **Dynamic Image script**: Provided that its selector refers to an image, this script dynamically changes the image for each record. See "Dynamic images" on

page 366.

- **Dynamic Background Script** (Print context only): This type of script dynamically changes the background of a Print section. See "Dynamic Print section backgrounds" on page 368.
- **Email scripts** (Email sections only): Email scripts define the sender, recipients, subject, dynamic attachments etc. of the email that is sent, and the PDF password. See "Email header settings" on page 180.
- **Conditional Content Script**: This script can conditionally show or hide any element in the template. See "Showing content conditionally" on page 358 and "Conditional Content script dialog" on page 539.
- **Conditional Print Section Script** (Print context only): This script conditionally shows or hides any Print section in a template. See "Conditional Print sections" on page 363.

- **Collapse All**: Collapses all the folders, hiding the scripts inside of them.

Scripts Pane filter

To quickly find a certain script or piece of code, type a search text in the filter field located below the buttons in the Scripts pane.
The Scripts Pane will then only display scripts of which the name, selector or content matches the search text. This is case insensitive.

Scripts Pane column

- **Name**: The name added to better identify the script.
- **Selector**: Displays the initial text or selector that the script applies to. See "Selectors in Connect" on page 414. Control Scripts and Post Pagination Scripts do not have a selector (see "Control Scripts" on page 429 and "Post Pagination Scripts" on page 445).

Tip

Fields from the Data Model pane can be dragged directly into the Scripts pane to create a Text Script. Additionally, Text scripts can be dragged into any section to add the script's placeholder at the insert location. See "Variable Data" on page 351.

Contextual menu options

- **New**: Click to create a new script or folder. See "Buttons" on page 591 for a description of all options. Which script types are available depends on the current section and script folder.
- **Duplicate**: Click to create an exact copy of the script.
- **Delete**: Click to delete the selected script. This does not delete any element or text in the template itself.
- **Rename**: Click to open a dialog to rename the script. This is the same as changing the **Name** field in the Edit Script window, which can be opened by double-clicking the script.
- **Enable/Disable**: Click to trigger the script to be enabled or disabled. Disabled scripts are greyed out and italic and will not be executed. See "Enable/disable scripts" on page 402
- **Import**: load a script from a Scripts file (*.OL-script).
- **Export**: save the script to a Scripts file (*.OL-script).
- **Properties** (Script folders only): edit the **name** and **execution scope** of the folder. See "Execution scope" on page 401.

Styles pane

The Styles pane shows which CSS style rules apply to the currently selected element.
A link next to a style rule will open the file where that particular style is defined. This can be either a CSS file or the source file of a section if local formatting was used (see "Styling and formatting" on page 294).

A crossed-out style rule signals that it was overruled by another style rule. This happens when:

- A more specific, and therefore more important rule, is encountered for the same element. See "Using a more specific CSS rule" on page 305 to learn more about the specificity of style rules.
- A rule with the same importance is read after the first rule. Not only is the order of the rules in a CSS file important, but also the order in which the style sheets are read. The style sheets that are included with a section are read in the specified order; see "Applying a style sheet to a section" on page 122.

Translations pane

The Translation pane contains translation entries. They can be exported to a POT file, which can be opened in a translation tool (see "Translating templates" on page 467).

Tip

Hover over a translation entry to see which HTML elements in the template have the same text and are tagged for translation (see "Tagging elements for translation" on page 469).

Buttons

- **Tag Element...**: Tag the currently selected element in the template for translation (see "Tagging elements for translation" on page 469). If no translation entry with the same text already exists, the text of the tagged element is copied to a new translation entry.
- **Add Empty String**: Adds a new, empty translation entry to the list and opens the "Translation String Options dialog" on page 560.
- **Export**: Exports the translation entries to a POT file (see "Exporting and importing translation files" on page 474).
- **Highlight Tagged Elements**: Turn highlighting on or off. Highlighting shows which HTML elements have the same text and are tagged for translation when you hover over a translation entry.
- **Sync**: Scans the template (all sections and snippets) for elements that are marked for translation and compares them to the list of translation entries. The results are shown in the "Sync Summary dialog" on page 552.
- **Edit locale**: Opens the "Locale Settings" on page 515 dialog.

Columns

- **Source**: The text that will be presented to the translator, and that will be replaced with a translation when OL Connect generates output from the template.
- **Context**: Describes where the source text is used, to help the translator translate the source text correctly in case it should be translated differently in different locations (see "Same text, different translation: add a context" on page 470).

Contextual menu options

- **Edit String**: Opens the "Translation String Options dialog" on page 560 for the clicked translation entry.

- **Delete**: Deletes the clicked translation entry and (optionally) removes the translation tag from matching elements.

Workspace

The Workspace pane is where everything comes together. It is the contents of the page, the WYSIWYG editor that shows what the output will look like.

The Workspace contains three tabs. To switch between the tabs, click on the tab at the bottom, or select **View > Design View**, **Preview View** or **Source View** on the menu.

A lock icon displayed next to the file name indicates that the file is read-only. Saving a read-only file opens the Save as dialog, allowing you to save the file under a different name. The new file is then opened and can be edited in the Designer.

For an overview of keyboard shortcuts, see "Keyboard shortcuts" on page 561.

Design tab

The Design tab shows the template including all styles, text and images as well as the placeholders used for variable data. In this tab, the template's scripts are not executed and only placeholders are shown.

Normally you would only edit a template in Design mode.

The top of the Design tab contains an area with the following options and buttons:

- **Breadcrumbs**: Displays the element type where the cursor is located and any of its parent elements. Elements with classes or IDs show these details next to them, for instance `div #contents > ol.salesitems > li`. Click on an element in the Breadcrumbs to select it.
  When an element is selected in the breadcrumbs and the Backspace or Delete key is pressed, that element is deleted.
  If the deleted element was targeted by a script, you will be asked if you want to delete the script as well.
- **Context Selector**: Displays the current context. The drop-down lists available contexts. Clicking on a context switches to that context.
- **Section Selector**: Displays the currently active section. Clicking on another section switches to that section.

- **Media Selector** (Master Page editor only): Displays a list of Media resources. Clicking on a media will display its Virtual Stationery background while in Preview mode.
- **Zoom Level**: Displays the current zoom level and drops-down to change the level.
- **Buttons**
  - **Zoom in**: Zooms in by 25%
  - **Zoom out**: Zooms out by 25%
  - **Actual Size**: Zooms to 100%.
  - **Fit Width**: Adjusts zoom to fit the exact width of the template to the available workspace.
  - **Refresh**: Reloads the view, including static external images and remote stylesheets, and re-runs the scripts (the latter in Preview Mode only).
  - **Responsive Design View**: Use the drop-down to select a specific screen width, to test the design for different devices. Not available in Print contexts.

> **Tip**
>
> You can also use the mouse's scroll wheel in combination with the Ctrl button to gradually zoom in or out.

When a template is open, the workspace also shows a ruler and guides (see "Guides" on page 314).

Preview tab

The Preview tab shows the section as it will output with the current record (see "Loading data" on page 341), showing the personalized content (see "Personalizing content" on page 338). Content added by a script isn't visible in the Design tab, but is visible and can be inspected in the Preview mode.
Although it is sometimes possible to edit the template in Preview mode to a certain extent, it is recommended to do all editing in Design mode.

Source tab

The source tab displays the HTML source of the section, including HTML Headers, CSS and HTML code. The source is displayed in a color-coded text editor, to quickly visualize the code. In this tab changes and adjustments can be made to the code. When editing a template in

Source mode you should write HTML. For shortcuts that can be used in this editor, see "Text editors: Source tab, JavaScript, CSS, Script Editor" on page 566 and "Emmet" on page 169.

To the left of the Source tab, a bar helps visually identify the start and stop of an element. For example when clicking on the opening <table> element, this bar marks the whole <table> and all its contents, until the ending </table> tag.

Pretty print options

In the Source view the HTML source of the template is "pretty printed" (that is: formatted, adding new lines and indentation) to make it more readable.
When this is undesirable, the Source view formatting can be turned off for (parts of) a section by adding <!-- format:off --> in Source view, at the beginning of the text or in between two HTML elements. From that point on, pretty printing will be disabled for that section. Use <!-- format:smart --> or <!-- format:auto --> to turn the formatting back on.

These are all format options:

- <!-- format:off --> turns the formatting off
- <!-- format:smart --> or <!-- format:auto --> turns the formatting back on.
- <!-- format:collapse --> forces the following HTML elements to be collapsed.
- <!-- format:expand --> forces the following HTML elements to be expanded.

Example

This example shows how to turn pretty printing off for one specific HTML element: a Div element.

```
<p>In a "pretty printed" paragraph
        <br>text is indented automatically on the Source tab.
</p>
<!-- format:off --><div anchor="page_media_0" style="font-family:
Lucida Console,monospace; line-height: 1.2; white-space: pre-wrap;
position: absolute; overflow: hidden; -moz-box-sizing: border-box;
width: 675px; height: 68px; top: 209.967px; left: 65.4px;" offset-
x="103.19999694824219" offset-y="247.76666259765625">@Header1@
</div><!-- format:auto-->
```

Toolbars

This topic lists the buttons that are available in the top toolbar of the Designer module.
For a description of the buttons at the top of the Workspace, see "Workspace" on page 595.

File manipulation

- **New**: Displays the New Wizard where a new data mapping configuration or a new template can be created.
- **Open**: Displays the Open dialog to open an existing template.
- **Save**: Saves the current template. If the template has never been saved, the Save As... dialog is displayed.
- **Print**: Opens the Print Output dialog.
- **Proof Print**: Opens the "Print options" on page 704 dialog as a "Proof Print" which limits the number of records output. The options themselves are identical to the regular Print Output dialog.

Output

- **Send Email**: Opens the Send Email dialog; see "Send (Test) Email" on page 543.
- **Send Test Email**: Opens the Send Test Email dialog; see "Send (Test) Email" on page 543.
- **Preview HTML**: Opens the current template's Preview in the system default browser. Useful for testing scripts and HTML output.

Pagination (Print Context only)

- **Page Number**: Inserts a placeholder for the current page number
- **Page Count**: Inserts a placeholder for the total number of pages in the current section.

Guides

- **Insert Horizontal Guide**: Click to insert a new horizontal guide; see "How to position elements" on page 312.

- **Insert Vertical Guide**: Click to insert a new horizontal guide; see "How to position elements" on page 312.

Form Wizard

- **Form Wizard**: Click to open the Form Wizard to add a form to a Web Context. See Forms and Form Elements.
- **Validation Settings**: Click to open the Validation settings dialog to change the validation settings on the currently selecting tools. See Forms.

Table manipulation

- **Insert Standard Table...**: Inserts a table with a specific number of columns and rows through the "Table" on page 278 Wizard.
- **Select**
  - **Select Table**: Selects the table where the cursor is located. If the cursor is within a table embedded within another, the innermost table is the one selected.
  - **Select Row**: Selects the innermost row where the cursor is located.
  - **Select Cell**: Selects the innermost cell where the cursor is located.
- **Delete**
  - **Delete Table**: Deletes the innermost table where the cursor is located.
  - **Delete Row**: Deletes the innermost row where the cursor is located.
  - **Delete Column**: Deletes the innermost cell where the cursor is located.
- **Insert**
  - **Insert Row Above**: Inserts a row above the current one. The row configuration, such as merged cells and cell styles, are duplicated, but contents is not.
  - **Insert Row Below**: Inserts a row below the current one. The row configuration, such as merged cells and cell styles, are duplicated, but contents is not.
  - **Insert Column Before**: Inserts a column to the left of the current one. The column configuration, such as merged cells and cell styles, are duplicated, but contents is not.

- **Insert Column After**: Inserts a column to the right of the current one. The column configuration, such as merged cells and cell styles, are duplicated, but contents is not.

Objects

- **Insert Image...**: Inserts an Image using a resource that is local to the template, at the current location of the pointer and opens its properties. See "Images" on page 271.
- **Insert Barcode**: Displays a list of available barcodes. Click on one to insert it on the page. See "Barcode" on page 197.
- **Insert Pie Chart**: Click to insert a new Pie Chart object and open the Chart Script wizard.
- **Insert Bar Chart**: Click to insert a new Bar Chart object and open the Chart Script wizard.
- **Insert Line Chart**: Click to insert a new Line Chart object and open the Chart Script wizard.

Hyperlinks

- **Insert Hyperlink...**: Creates a Hyperlink or mailto link on the currently selected text or element and opens its properties. See "Hyperlink and mailto link" on page 268.
- **Remove Hyperlink**: Removes the currently selected hyperlink. The text or element that was the hyperlink is not removed.

Boxes

- **Insert Positioned Box**: Inserts an absolute-positioned box on the page, which can be moved around freely.
- **Insert Inline Box**: Inserts an inline box that is set to float to the left, at the position of the cursor.
- **Wrap in Box**: Takes the current selection and wraps it inside a new box.
- **Float Left**: Floats the current element to the left using a *float:left* style.
- **No Float**: Removes any *float* style applied to the currently selected element.
- **Float Right**: Floats the current element to the right using a *float:right* style.

- **Rotate Counter Clockwise**: Rotates the currently selected box 90° counter-clockwise.
- **Rotate Clockwise**: Rotates the currently selected box 90° counter-clockwise.

Align Objects

- **Top**: Aligns the top side of the selected objects with the top edge of the last selected object.
- **Middle**: Aligns the middle of the selected objects with the middle of the last selected object. Objects may move up or down.
- **Bottom**: Aligns the bottom side of the selected objects with the bottom edge of the last selected object.
- **Left**: Aligns the left side of the selected objects with the left edge of the last selected object.
- **Center**: Aligns the center of the selected objects with the vertical center of the last selected object. Objects may move to the left or to the right.
- **Right**: Aligns the right side of the selected objects with the right edge of the last selected object.

Styles

- **Element Type**: Displays the element type of the selected element and drops down to show other element types in which it can be changed.
- **Style**: Displays the style of the selected element and drops down to show other available styles which can be applied to it.
- **Font Face**: Displays the font face of the selected text or element where the cursor is located and drops down to show other available font faces which can be applied to it. Fonts added to the Fonts folder of the Resources pane are shown automatically in the Fonts drop-down.
- **Font Size**: Displays the font size of the selected text or element where the cursor is located and drops down to show other available sizes which can be applied to it.
- **Font Color**: When text is selected, click to apply the shown color to the selected text, or use the drop-down to change the color and apply it.

Alignment

- **Align Left**: Aligns the currently selected element to the left.
- **Align Center**: Aligns the currently selected element to the center.
- **Align Right**: Aligns the currently selected element to the right.
- **Justify**: Aligns the currently selected element to stretch text lines to fill all available width.

Text editing

- **Bold**: Makes the currently selected text **bold**.
- **Italic**: Makes the currently selected text *italic*.
- **Underline**: Makes the currently selected text underline.
- **Strikethrough**: Makes the currently selected text ~~strikethrough~~.
- **Create Numbered List**: Makes the selected text element a numbered list (<ol>). If multiple paragraphs are selected, each becomes a list item (<li class="Bullet">).
- **Create Bulleted List**: Makes the selected text element a bullet list (<ul>). If multiple paragraphs are selected, each becomes a list item (<li class="Bullet">).
- **Indent**: Increases indentation of the selected text element. If the element is a paragraph, it is wrapped in a <blockquote> element. If it is a list item, it is moved to a child level, creating a new list if necessary.
- **Outdent**: Decreases indentation of the selected text element. If the element is wrapped in a blockquote element, one blockquote is removed. If the element is a list item, it is removed from one surrounding list.
- **Superscript**: Makes the currently selected text a superscript.
- **Subscript**: Makes the currently selected text a subscript.
- **Remove Formatting**: Remove any and all styles, text decorations and other formatting from the selected text. Indentation is not affected.

Miscellaneous

- **Insert Lorem Ipsum**: Inserts a paragraph of generic lorem ipsum text, useful for placeholder or template design.
- **Show Edges**: Shows a colored border around elements on the page and the type of element that is highlighted.
- **Welcome Screen**: Click to re-open the Welcome Screen.

Welcome Screen

The **Welcome Screen** appears when first starting up PrintShop Mail Connect. It offers some useful shortcuts to resources and to recent documents and data mapping configurations.

If you are new to PrintShop Mail Connect and you don't know where to start, see "Welcome to PrintShop Mail Connect 2022.1" on page 12.

The Welcome Screen can be reopened in two ways:

- The **Welcome Screen** button in the "Toolbars" on page 598.
- From the Menus in **Help**, **Welcome Screen**.

To go back from the Welcome Screen to the template or data mapping configuration that you were working on:

- Close the Welcome Screen by clicking the cross next to the text 'Welcome' at the top.

Contents

- **Resources**
  - **Documentation**: Opens this documentation.
  - **Training**: Opens **Learn**: the Objectif Lune e-Learning Center, with its tutorials, forum, and how-tos.
  - **Support**: Opens the support page in the PrintShop Mail Connect website.
  - **Licenses & Activations**: Opens the **Objectif Lune Web Activation Manager**.
  - **License**: Shows your current license's details.
  - **Website**: Opens the PrintShop Mail Connect website.

- **More options**
  - **Get Printer Configurations (PDC)**: Lists printer definition files available for download. The selected files will be downloaded to the current user's Connect\workspace\configurations\ folder and can be imported via the Print Wizard; see "Adding print output Models to the Print Wizard" on page 1005.
  - **Get Finishing Configurations (HCF)**: Lists HCF Files available for download. High Capacity Feeders (HCF) are also commonly referred to as Inserters or Folder-Inserters. The Inserter options in the print settings are dependent upon the HCF model selected; see "Inserter options" on page 783.
- **Take A Tour**: Click to open the YouTube Playlist giving you a tour of the software.
- **Create/Open**:
  - **Open File**: Lets you open an existing template or data mapping configuration.
  - **New template**: Lets you choose a Context to create a new template without a Wizard.
  - **Template Wizards**: Displays a list of available Template Wizards, producing pre-made templates with existing demo content; see "Creating a template" on page 109.
  - **Sample Projects**: Displays a list of available Sample Projects, producing a complete Connect project; see Sample Projects.

Print options

The **Print Options** page is the first page of the **Advanced Print Wizard** (used for both Production and Proof printing) and the second page of the "Output Creation Presets Wizard" on page 700.

This page is the most important of both the Advanced Print Wizard and the ***Output Creation Preset*** Wizard. All other pages that appear in the Wizard are determined by the selections made on this page.

The options that don't appear in the ***Output Creation Preset*** Wizard (as noted below), do appear in the ***Job Creation Preset*** Wizard. Only the Advanced Print Wizard contains all the options, but it cannot save your settings to file like the preset wizards allow.
For more information about print preset files, see "Print Presets" on page 998.

The choices can be broken down as follows.

Printer

- **Model**: Use the drop-down to select the printer language / output type that will be generated.
  Connect comes with several bundled print output Models which cover a large range of industry standard print output types.
  These include AFP, IPDS, PCL, PDF and PostScript (including the optimized PPML, VIPP and VPS variants), with a range of quality settings available for each.

  The bundled print output Models do not cater for *all* types of printers, however. There are simply too many different printers with their own individual (often quirky) settings and capabilities to cater for all of them with just a few standardized print Models. To use Connect with a particular printer Model you must obtain a tailored Connect Printer Definition that defines that printer and its capabilities for Connect. For example, to cater for IPDS or AFP printers that support Encapsulated Postscript resources included through an Include Object (IOB) structured field. Such a print output Model would need to be customized for you. These customized Printer Definitions will be created by Upland OL Support, in conjunction with information you provide about both your printer and your specific needs. Support will then provide you with a customized Printer Definition that supports the desired functionality with your specific printer.

  By default, Connect initially displays just a single generic *PDF output* option, to keep the interface clean. But other print output types can be added to the Printer Model drop-down list at any time via the **Import Definitions** button

  The **Import Definitions** options are:

  - **Import Definition**: Used to add any customized or tailored Printer Definition that Upland Objectif Lune has prepared for you, specific to your printer.
  - **Create Definition from PPD**: Used to create your own customized PostScript Printer Model through the printer manufacturer provided PostScript Printer Definition (PPD) file. This option launches the "Dynamic PPD Options" on page 788 page later in the Printer Wizard, which allows you to setup the printer.
  - **Edit available printers**: Add extra Printer Definitions from the selection that come installed with Connect.
  - **Export printer definition from output preset**: Use to extract a Printer Definition from a new or modified Printer Model. This option is only available when the Printer Model currently loaded differs from those previously saved or installed.

  For more information on how to add Printer Definitions to Connect, see "Adding print

output Models to the Print Wizard" on page 1005.

Output options

- **Output Local**: Select to have the output created using the local Print Server.
- **Output Type** choices:
  - **Prompt for file name**: Select to output to a local file on the hard drive. When this option is selected, no other configuration is necessary. A Save As dialog will appear to allow selection of the folder and file name.
  - **Directory**: Select to output to a local folder on the machine.
    Selecting this will open the **Directory Options** sub-group, which has these options:
    - **Job Output Mask**: The name of the file that will output.
      You could write the Job Output Mask directly into this edit box (for a list of available variables see "Print output variables" on page 1008), or you could create a Mask via the Options button. This opens the custom dialog: Job Output Mask Dialog.
      The Job Output Mask may contain (dynamic) folder names, for example:
      `${document.metadata['Country']}\${template}.`
      The evaluated value of the Job Output Mask is taken as a path **relative** to the folder specified by the Job Output Folder (the next option in this dialog). The Job Output Folder must exist, but folders specified in the Job Output Mask will get created if they don't exist.
    - **Job Output Folder**: The path on the disk where the file is produced. Please note that the folder must exist, or output will fail when produced through the server.
  - **LPR Queue**: Select to send the print job to an LPR queue. It is assumed that the print technology is supported by the system receiving the LPR job.
    - **Local Printer**: The IP or host name of the printer or machine where the LPD is installed and will receive
    - **Queue Name**: The queue name that will accept the job on the LPD. Default is generally "auto".
    - **Job Owner Name**: Optional entry for adding the name of the job owner.
    - **Job Name**: The name of the output file. You can use ${template} as a variable for the name of the Designer Template used to generate the output. (See also: "Print output variables" on page 1008.)

- **Windows Printer**: Select to send the Print Job to a printer queue (note that this is an actual printer queue, not a Workflow Printer Queue).
  For non-PDF output, the job will be first rendered as a PDF before being printed through the Windows driver.
  For PDF output the Adobe Library is used.
  - **Windows Printer**: Use the drop-down to select the Windows printer queue where the job will be sent.
  - **Job Owner Name**: Optional entry for adding the name of the job owner.
  - **Job Name**: The name of the output file. You can use ${template} as a variable for the name of the Designer Template used to generate the output.
  - **PDF Rendering Options** sub-group (only available for *PDF output*):
    - **Auto-rotate and center**: Check to automatically select the page orientation that best matches the content and paper.
    - **Choose paper source by page size**: Check to use the PDF page size to determine the output tray rather than the page setup option. This option is useful for printing PDFs that contain multiple page sizes on printers that have different-sized output trays.
    - **Scale**: Chose from one of the following scaling options:
      - **None**: Select to not scale any page, whether it fits or not.
      - **Expand to printable area**: Select to expand any page to fit the page area. Pages larger than the paper size are not re-sized.
      - **Shrink to printable area**: Select to shrink any page to fit the page area. Pages smaller than the paper size are not re-sized.

Production Options

- **Booklet Imposition**: Check to tell the printer to generate a booklet for the print output. Booklet options are set in the "Booklet Options" on page 773 page.
  This option is unselected by default unless selected in the Designer "Print section properties" on page 542.
- **Imposition**: Check to enable Cut & Stack Imposition, which is set in the "Imposition options" on page 774 page.
- **Add Inserter marks**: Check to enable Inserter mark functionality, which is set in the "Inserter options" on page 783 page.

- **Runtime Parameters**: select "Runtime Parameter Options" on page 686 to store information which can be used at runtime for comparisons against conditions within Job Creation, or for use with external sorting programs.
- **Override Finishing options** (not available in ***Output Creation Preset*** Wizard): Check to configure custom "Finishing options" on page 691, such as binding.
- **Print virtual stationery**: Check to enable virtual stationery in the output.
- **Use grouping** (not available in ***Output Creation Preset*** Wizard): Check to configure grouping of output into jobs, job segments or document sets. See "Grouping options" on page 696.
- **Include meta data** (not available in ***Output Preset Creation*** Wizard): Check to add meta data to the output. This can be done at Job, Job Segment, Document, Document Set and Page level. See "Meta Data options" on page 699.
- **Separation**: Check to activate the "Separation options" on page 807 page of the wizard.
- **Add additional content**: Check to activate the "Additional Content" on page 732 page of the wizard.

Records

(Not available in ***Output Creation Preset*** Wizard.)

- **Record Range**: Allows selection of a range of records or a custom selection.
  You can specify individual records separated by semi-colons (;) or ranges using dashes.
  For example: 2;4;6-10 would print pages 2, 4, 6, 7, 8, 9 and 10.
- **Apply filtering and sorting to record selection**: Check to activate the "Data filtering options" on page 689 (to filter out certain records) and "Sorting options" on page 693 (to sort the remaining records) pages of the wizard.

Copies

(Not available in ***Output Creation Preset*** Wizard.)

- **Copies**: Enter the number of copies to print, of each record.
- **Collate**: When printing multiple copies you can check this checkbox to have the record copies printed together.
  For example in a three record job the records would print out as 1-1-2-2-3-3, rather than 1-2-3-1-2-3.

Pure Color Thresholds

This section is valid for AFP, IPDS and PCL only. It applies to elements within the record that are shades of grey, rather than black or white.

- **Black Threshold Percentage**: The percentage of shading at which the element will appear as full black, rather than dark grey.
- **White Threshold Percentage**: The percentage at which the element will appear as full white, rather than light grey.

Advanced Print Wizard navigation options

- **Load** button (not available in ***Output Creation Preset*** Wizard): Click to select a previously created Output Creation Preset.
  This will change the Advanced Print Options to match the entries contained within the Preset.
- **Preview** button (active in ***Advanced Print*** Wizard only): Click to launch a Proof Preview window, which displays how the printed output would look based upon the currently chosen selections.
- **Back** and **Next** buttons: Used to navigate back and forth through all the selected options within the Wizard. Up until the Print button is pressed, one can reverse all the way through the wizard to return to the main selection page (the "Print options" on page 604 page) and add or remove printing options from the print run.
- **Print** button (***Advanced Print*** Wizard only) or **Finish** button (***Output Creation Preset*** Wizard only):
  Click to produce print output/finalize the Preset according to the current settings.
  This can be done at any point within the Wizard, whether or not *all* the options selected in the "Print options" on page 604 page have been completed or not.
- **Cancel** button: Cancels the Print Wizard, without creating any printout.

Booklet Options

This page appears as part of the **Advanced Print Wizard** and the Output Creation Preset Wizard.

The **Booklet Options** page defines how to generate booklets in the output. It is used in conjunction with Imposition settings, which will appear after the Booklet entries have been made.

This page includes a handy illustration that displays how the final binding would look, based upon the current selections.

Options:

- **Configuration**: Use the drop-down to select the type of binding to use:
  - **Saddle Binding**: This binding places all the pages in a stack, binds the middle and folds the stack as one.
  - **Perfect Binding**: This binding type is often used for books. Pages are folded in the middle and then set side by side. The pages are then bound along the folded "spine".
  - **1 up Perfect Binding**: This binding does not contain any folding. The pages are lined up side by side and bound along one edge.
- **Booklet Binding Edge**: Use the drop-down to select the side on which to bind the booklet.

Optional **Cover Page** selections are available to Saddle Binding only.

- **Cover Page** checkbox: Check to enable cover pages to be created with the options below:
  - **Media** selections:
    - **Cover Media Size**: Use the drop-down to select the media size for the cover page, or use a Custom size and select **Width** and **Height** values.
  - **Front Cover** selections:
    - **Blank**: Select to add no data to the front cover.
    - **First page on outside and second page on inside**: Select to use the first 2 pages as the inside and outside of the front cover.
  - **Back Cover** selections:
    - **Blank**: Select to add no data to the back cover.
    - **Last two pages on inside and outside**: Select to use the final 2 pages as the inside and outside of the back cover.

Advanced Print Wizard and Output Creation Preset Wizard navigation options

- **Preview** button (active in ***Advanced Print*** Wizard only): Click to launch a Proof Preview window, which displays how the printed output would look based upon the currently

chosen selections.

- **Back** and **Next** buttons: Used to navigate back and forth through all the selected options within the Wizard. Up until the Print button is pressed, one can reverse all the way through the wizard to return to the main selection page (the "Print options" on page 704 page) and add or remove printing options from the print run.
- **Print** button (***Advanced Print*** Wizard only) or **Finish** button (***Output Creation Preset*** Wizard only):
  Click to produce print output/finalize the Preset according to the current settings.
  This can be done at any point within the Wizard, whether or not *all* the options selected in the "Print options" on page 704 page have been completed or not.
- **Cancel** button: Cancels the Print Wizard, without creating any printout.

Imposition options

This page appears as part of the **Advanced Print Wizard** and the Output Creation Preset Wizard.

Imposition refers to the printing of multiple pages on a single sheet. This is also known as N-Up printing.

The options on the **Imposition Options** page allow for the setting of imposition repetition, order, margins and markings.

As imposition selections are very specific and can be quite confusing, we have provided a handy dynamic diagram that displays a representation of the current imposition selections in real time. Whenever selections are changed, this display changes to reflect the selections made.

You can even select how many pages are to be represented, whilst page numbers appear on the pages in the diagram and change dynamically, like everything else.

It is important to note that if Impositioning is used then the only meta data that will be available to the job thereafter will be the meta data at Job Segment level.

When using Impositioning everything within the Segment is Impositioned, so all meta data below the Job Segment level is lost. If *Booklet Binding* were selected, some of the imposition settings will be determined by the options made within the "Booklet Options" on page 773 Page and those imposition settings will thus be disabled on this page.

Note

During Impositioning, all hyperlinks - both external and internal - are removed from the document (see "Hyperlink and mailto link" on page 268).

The Imposition selections that can be made are as follows:

- **Sample Page** group: Use to choose a paper size for the logical page(s) which would be printed upon the actual physical Imposition sheet. This helps in decision making regarding what paper sizes and orientations to use, with the Sample Page shown in blue on the white Imposition Sheet in the output preview (and validation) diagram.

  Note

  The logical page size can be set independently of any template open in the Designer, allowing for output presets to be created independently from the active template.

  If a template is currently loaded in Connect Designer, the size and portrait attribute of the first section of that template are used as defaults for the Sample Page.

  As a general rule, it makes little sense to modify the Sample Page size within the **Print Wizard** (as the template media size will already have been selected), but it does make sense when creating Print Output Presets for other templates.

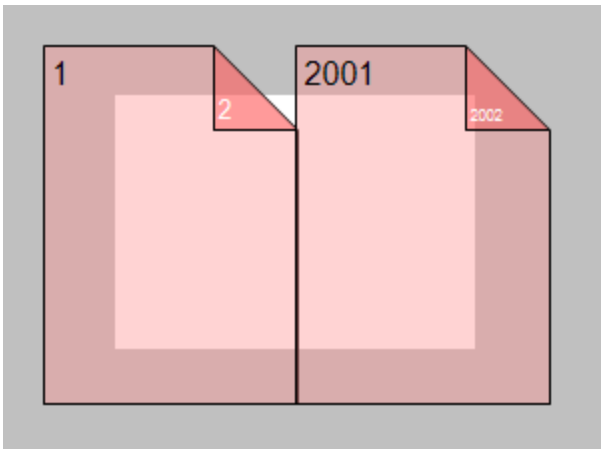
  If no template is loaded or if the template does not have a Print section, then the Sample Page defaults to A4 portrait.

  - **Size**: Use the drop-down to select the media size of the logical template page.
  - **Orientation**: Select orientation (aspect ratio) of the logical template page (*Landscape* or *Portrait*).
  - The **Width** and **Height** values are real-only and appear for display and design purposes only.
- **Imposition Sheet** group: Use to select the physical media size that the logical pages will be printed upon. This appears as the white background in the output preview (and validation) diagram.

- **Size**: Use the drop-down to select the size of the physical media the output is to be printed upon.
- **Orientation**: Select orientation (aspect ratio) of physical media sheet (*Landscape* or *Portrait*).
- The **Width** and **Height** values are real-only and appear for display and design purposes only.

- Output diagram and validation. Use the settings here to control how the output diagram is to look and behave. It contains the following options:
  - **Display size** entry: Chose between a pre-set viewing percentage, or manually enter whatever percentage you would like, or select *Fit Page* or *Fit Width* to have the diagram display the page(s) within the diagram window dimensions.
  - Use the zoom in/out buttons / to increase or reduce the viewing percentage by 25% increments.
  - **Sample size** entry: Set the sample output size here. The output diagram will change dynamically to reflect the selection made here, if applicable.
    The maximum Sample size is 10,000 for booklet imposition and 10,000,000 for standard impositions.
  - Output diagram: A real-time display that reflects the selections made in the Imposition Options page. The blue page(s) are the individual logical page(s), upon a white background that represents the physical Imposition Sheet.

    If the settings are invalid, the display will appear as red, such as in the following diagram:



In this example the error is that the logical pages are too large to fit upon the Imposition Sheet that has been selected.

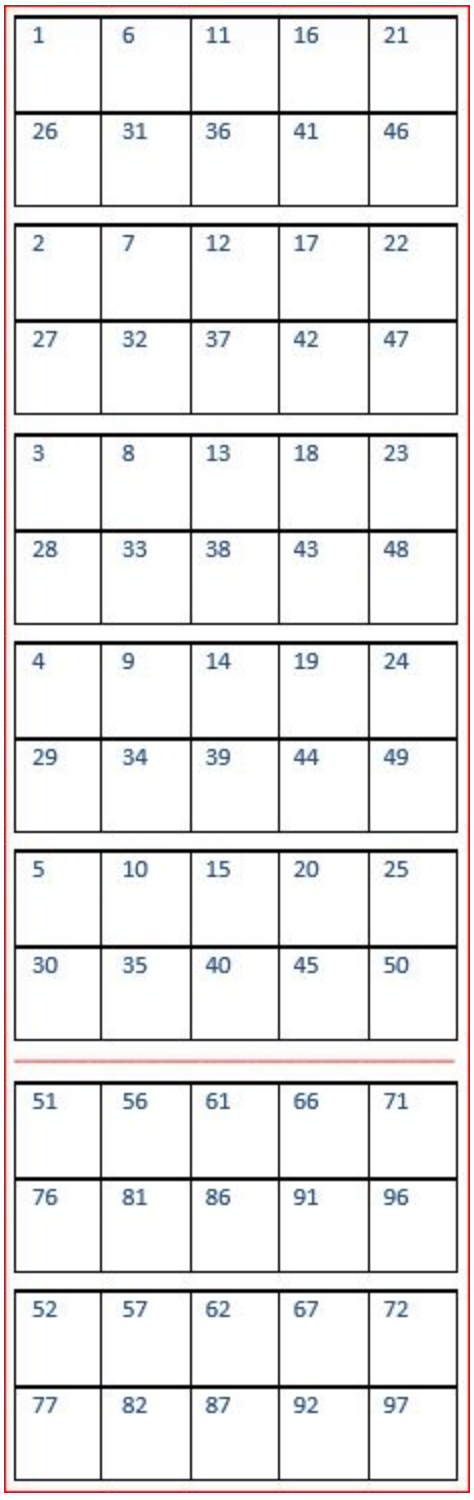
- **Position** group: These settings apply to the positioning of the logical page(s) within the Imposition Sheet.
  - **Method** selection: Set the position of the logical page(s) upon the Imposition Sheet. The choices are as follows:
    - *Center*: Have the logical page(s) centered within the Imposition Sheet.
    - *Scale to fit*: Scales the logical page(s) so that they fit the Imposition Sheet exactly.
    - *Offset*: Have the logical page(s) offset within the Imposition Sheet. The offset values can be set in the Left and Top entry boxes, which only become available for this positional method.
  - **Left** and **Top** values: These entry boxes are only enabled and their values applied when the Offset method has been chosen. They are used to set the offset of the logical page(s) upon the Imposition sheet.
  - **Rotate** selection: Select to rotate the logical page(s) within the Imposition Sheet.
- **Layout** group: Select how the logical page(s) are to be laid out on the Imposition Sheet.
  - **Horizontal** selection: Set how many logical pages are to be placed on one horizontal line within the Imposition Sheet.

    > **Note**
    >
    > If the logic page was set to **Rotate** in the Position selections, then the Horizontal selection might appear as vertical, and the Vertical selection might appear as horizontal, on the real time output diagram.

  - **Vertical** selection: Set how many logical pages are to be placed on one vertical line within the Imposition Sheet.
  - **Horizontal Gap** selection: Set the amount of blank space to add between each logical page on this axis.
  - **Vertical Gap** selection: Set the amount of blank space to add between each logical page on this axis.
  - **Stack Depth** selection: Stack Depth defines the number of Imposition Sheets that will be printed before drill sorting resets.

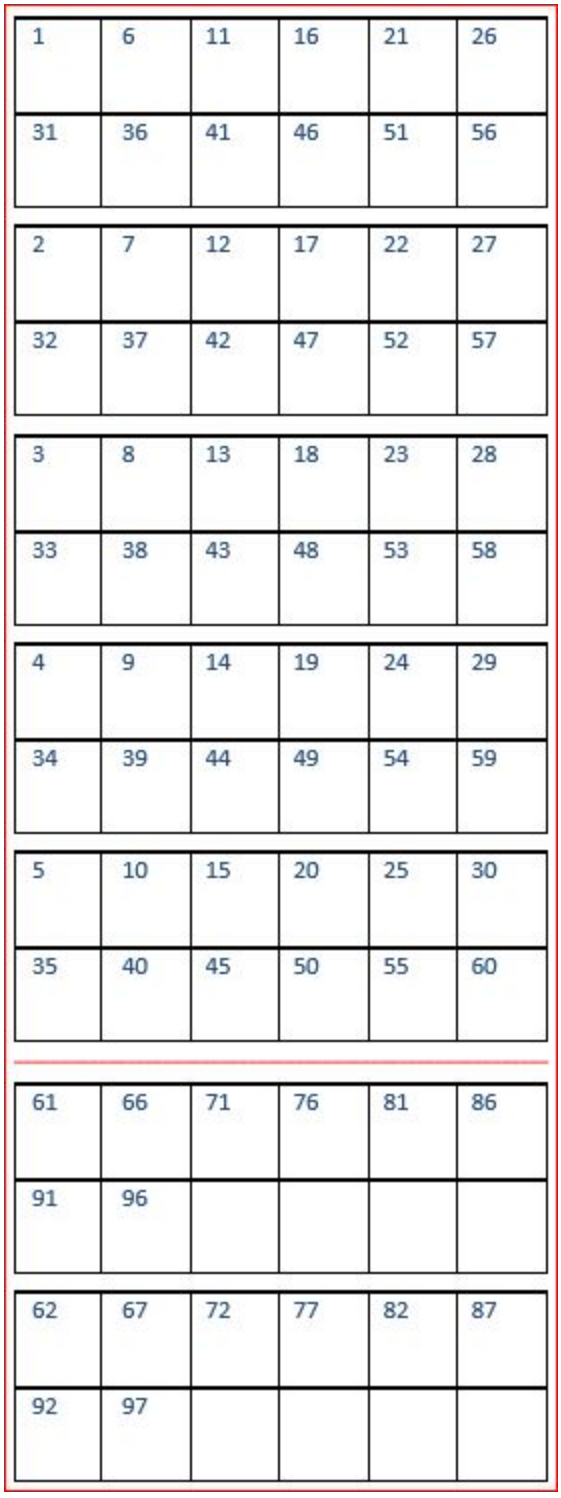
Stack Depth works in conjunction with the Horizontal and Vertical page, Gap and Page Order selections in order to determine the printing order. This is best illustrated by way of example.

The following "Cut and stack" example has a job with 100 single page documents and repetition settings of 5 Horizontal and 2 Vertical, plus a Stack Depth of 5. Each table represents a single Imposition Sheet, and the red line between the tables represents the Stack Depth:



Each set of 5 Imposition Sheets is a discrete drill sorted nUp block that can be guillotined and stacked. The first 5 sheets encompass records 1-50, whilst the next 5 sheets encompass records 51-100.

The following example is the same job but with a Horizontal repetition setting of 6:



1	6	11	16	21	26
31	36	41	46	51	56
2	7	12	17	22	27
32	37	42	47	52	57
3	8	13	18	23	28
33	38	43	48	53	58
4	9	14	19	24	29
34	39	44	49	54	59
5	10	15	20	25	30
35	40	45	50	55	60
61	66	71	76	81	86
91	96				
62	67	72	77	82	87
92	97				

As you can see, the outputs are *very* different.

Note

If the Stack Depth is greater than the number of sheets in the entire job then you will get a single drill sorted block that can be guillotined and stacked to give the complete job in record order.

- **Sides** selection: Select how many sides of the Imposition Sheet are to be printed upon. Choice is between *Simplex* or *Duplex*.
- **Method** selection: In 2018.2 we introduced a new imposition page order for cut & stack impositioning that works with continuous feed devices: *Stack by Column*. This option is intended for impositions with more than 1 row. It places the next page in the next row of the same column on the current sheet, instead of in the same row and column of the next sheet like the pre-existing *Cut and Stack* does.
  The choices are between:
  - *Cut and Stack*: Any preset created in a version of Connect prior to Connect 2018.2 will default to this Layout Method. It allows total control of subsequent **Page Order** choices.
  - *Stack by Column*: For use with continuous feed printers. This selection limits subsequent **Page Order** choices to column based options only.
- **Page Order**: Select in which direction to go when adding sections to the output:
  - **Left to right, top to bottom**
  - **Right to left, top to bottom**
  - **Top to bottom, left to right**
  - **Top to bottom, right to left**
- **Reverse** selection: The reverse selections have been extended in Connect 2018.2, with the addition of the *Inverse page order* option. The options now available are:

- *Off*. Output is not reversed.
- *Stack upside down*: Essentially turns the stack upside down by reversing the output imposition page sides. The last sheet becomes the first, and in the case of duplex imposition, the back sides become the front sides.
  For presets created in a version prior to Connect 2018.2, a "Reverse Pages" selection will apply this *Stack upside down* setting.
- *Inverse page order*: This impositions the output in the ***exact opposite order*** from what has been selected.
  For example, if the selected order was **Top to bottom, left to right** (down the stack, left to right, top to bottom), then selecting **Inverse page order** would result in output that was up the stack, right to left, bottom to top.
  In the case of duplex impositioning, the front and back sides don't switch.

- **CropMarks** group: Select to add Crop Marks to the printed output.
  - **Type**: Use the drop-down to select whether to add Crop Marks to the page or not, and what type of Crop Marks to use. The Crop Mark types available are *Standard* or *Japanese*.
  - **Page side**: What side(s) of the page to put the Crop Marks. The choices are *Front*, *Back* or *Both*.
  - **Width**: Select the width of the crop mark lines.
  - **Offset**: How much separation (if any) to leave between the vertical and horizontal corner markings.
  - **Length**: Select the Length of the crop mark lines.

Advanced Print Wizard and Output Creation Preset Wizard navigation options

- **Preview** button (active in ***Advanced Print*** Wizard only): Click to launch a Proof Preview window, which displays how the printed output would look based upon the currently chosen selections.
- **Back** and **Next** buttons: Used to navigate back and forth through all the selected options within the Wizard. Up until the Print button is pressed, one can reverse all the way through the wizard to return to the main selection page (the "Print options" on page 704 page) and add or remove printing options from the print run.
- **Print** button (***Advanced Print*** Wizard only) or **Finish** button (***Output Creation Preset*** Wizard only):
  Click to produce print output/finalize the Preset according to the current settings.

This can be done at any point within the Wizard, whether or not *all* the options selected in the "Print options" on page 704 page have been completed or not.

- **Cancel** button: Cancels the Print Wizard, without creating any printout.

Inserter options

This page appears as part of the **Advanced Print Wizard** and the Output Creation Preset Wizard.

The **Inserter Options** page allows the selection of a High Capacity Feeder (HCF) model. These machines are also commonly referred to as Inserters or Folder-Inserters.

The options available on this page are dependent upon the model selected.

The options selected on this screen influence the position of the markings set on the next page: **"Mark Position Options" on page 785**.

- **Model**: Use the drop-down to select from any previously loaded Inserter model, or use the Browse button to select a HCF file to load a new Inserter model.
  An image representing the chosen folder-inserter is displayed under the list, along with the HCF file details.
- **Options Group:**
  The options available here are all Inserter dependent, and thus will change based upon the Inserter model selection.
  To see how the selected Inserter markings would look on the printed page, click the Next button to move to the "Mark Position Options" on page 785 page, which has a preview of the page. You can move back and forward between these two pages until you are entirely satisfied with the selections made.
  - **Mark Configuration**: Use the drop-down to select the type of markings to add. This selection basically equates to the amount of area the markings will take up on the printed page.
  - **Fold Type**: Use the drop-down to select the type of fold to apply to the paper. This will impact upon where on the page the markings will be placed.
  - **Collation level**: Select whether the markings will be made at Document level, or Document Set level.
  - **Print marks on back**: Check to place the Inserter Marks on the rear of the page.
  - **Selective Inserts**: If selective inserts are supported by the chosen Mark Configuration you can select what markings to include and whether those markings

are to included based upon some conditional setting. This can be done by highlighting the Mark Name entry and either pressing the **Edit** button, or using the right mouse click context menu, and selecting **Edit**.
For information on how to edit the Selective Inserts settings, see "Selective Insert Dialog" on page 786

- **Clear Background Area** Tab:
  Check the **Clear Background Area** checkbox to add a white background to the OMR, preventing background colors or elements interfering with the OMR Markings when they are read by the Inserter.
  - **Margins**:
    - **Same for all sides**: Check so that the Left margin selection is used to set all sides identically.
    - **Left, top, right, bottom**: Enter measurements for the margins on each side of the OMR Marks.
- **Custom OMR mark sizing** Tab:
  If supported by the currently chosen Mark Configuration you can select a Custom OMR size by checking the **Custom OMR mark sizing** checkbox.
  Select from any of the following, or leave the entries blank to use default values:
  - **Line length**: Enter a value between 10.16mm and 20mm.
  - **Line thickness**: Enter a value between 0.254mm and 0.63mm.
  - **Gap distance**: Enter a millimeter value 2.91mm and 4.2mm.

Advanced Print Wizard and Output Creation Preset Wizard navigation options

- **Preview** button (active in ***Advanced Print*** Wizard only): Click to launch a Proof Preview window, which displays how the printed output would look based upon the currently chosen selections.
- **Back** and **Next** buttons: Used to navigate back and forth through all the selected options within the Wizard. Up until the Print button is pressed, one can reverse all the way through the wizard to return to the main selection page (the "Print options" on page 704 page) and add or remove printing options from the print run.
- **Print** button (***Advanced Print*** Wizard only) or **Finish** button (***Output Creation Preset*** Wizard only):
  Click to produce print output/finalize the Preset according to the current settings.
  This can be done at any point within the Wizard, whether or not *all* the options selected in

the "Print options" on page 704 page have been completed or not.

- **Cancel** button: Cancels the Print Wizard, without creating any printout.

Mark Position Options

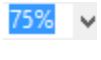
This page appears as part of the **Advanced Print Wizard** and the Output Creation Preset Wizard.

The **Mark Position Options** page displays a Preview of the output and the possible locations to place the inserter marks. The initial settings are determined by the selections made within the "Inserter options" on page 783 page.
You can move back and forward between these two pages to perfect the settings, or you could move the inserter mark box to the desired location on the preview.

Preview box:

- The *pink area* displays the areas of the page where inserter marks can be positioned.
- The *small checkered box* displays the current location of the inserter marks. This box is selectable and can be dragged to the desired location within the printable (pink) areas. If the box is placed outside the printable areas the page will display an error and prevent attempts at leaving the page.

Below the Preview box are buttons which allow control of the Preview box. The selections that can be made are:

- **First Page**: Click to jump to the first page.
- **Previous Page**: Click to move to the previous page.
- **Next Page**: Click to move to the next page.
- **Last Page**: Click to jump to the last page.
- **Show Page**: Use the up and down arrows or type a page number to display a specific page within the document.
- **Zoom in/out**: Click to zoom in or out by 25%
- 

Zoom Level: Use the drop-down to select a predefined level or enter a zooming percentage.

Selective Insert Dialog

The Selective Insert Dialog provides control over individual inserts. In this dialog the selected can be set to be either selected ("Yes") or not ("No"), or selected based upon some conditional criteria ("Conditional").

For example, you could add a marking to the third page of a document by making the selection Conditional and then setting the Condition entry to "page.nr == 3".

Both Metadata and Informational fields can be used in the Conditionals settings.

For more details on how to create a conditional, see the "How to Set Conditionals" on page 772 page.

Separation options

This page appears as part of the **Advanced Print Wizard** and the Output Creation Preset Wizard.

The **Separation Options** page defines how to separate a job into discreet portions, using either predetermined counts (of *Sheets*, *Documents*, *Document Sets* or *Job Segments*), slip sheets, or jogging.

- **Separate by Count** group:
  This group allows for the splitting of output based upon a specified number of *Sheets*, *Documents*, *Document Sets* or *Job Segments*.
  - **Split**: Use the drop-down to select whether or not to split the job based on number of *Sheets*, *Documents*, *Document Sets* or *Job Segments*.
    - **None**: Select this option to ignore count splitting.
    - **At Exactly**: Select to create a split based upon a predetermined number of *Sheets*, *Documents*, *Document Sets* or *Job Segments*.
  - **Every**: Enter the number of *Sheets*, *Documents*, *Document Sets* or *Job Segments* at which the output is to be split.
    This option is only available if the **Split** entry has been set to *"At Exactly"*.
  - Use the drop-down box to chose what the separation option is to be.
    The selection is either **Sheets**, **Documents**, **Document Sets** or **Job Segments**.
    The default separation option is *Sheets*, which was the only option available for splitting prior to Connect 2019.1.

- **Separation Settings** group:
  This setting is only available if no **Separate by Count** split has been specified.
  - **Separation**: Use the drop-down to select when a job separation occurs, which is either **None** (no separation) or at the **Job**, **Job Segment**, **Document** or **Document Set** level.
- **Slip Sheets** group:
  - **Add slip sheet**: Use the drop-down to select whether to add a slip sheet before or after a specific separation, or whether to use none.
  - **Every**: Use the drop-down to select at which separation to add a slip sheet, at the **Job**, **Job Segment**, **Document** or **Document Set** level.
  - **Media Size**: Use the drop-down to select the media size of the slip sheet.
    If a custom Media Size was chosen:
    - **Width**: enter slip sheet page width.
    - **Height**: enter slip sheet page height.
- **Jog** group:
  - **Jog after every**: Use the drop-down to select when to jog the printer, which is either **None** (no forced jogging) or at the **Job**, **Job Segment**, **Document** or **Document Set** level.

Advanced Print Wizard and Output Creation Preset Wizard navigation options

- **Preview** button (active in ***Advanced Print*** Wizard only): Click to launch a Proof Preview window, which displays how the printed output would look based upon the currently chosen selections.
- **Back** and **Next** buttons: Used to navigate back and forth through all the selected options within the Wizard. Up until the Print button is pressed, one can reverse all the way through the wizard to return to the main selection page (the "Print options" on page 704 page) and add or remove printing options from the print run.
- **Print** button (***Advanced Print*** Wizard only) or **Finish** button (***Output Creation Preset*** Wizard only):
  Click to produce print output/finalize the Preset according to the current settings.
  This can be done at any point within the Wizard, whether or not *all* the options selected in the "Print options" on page 704 page have been completed or not.
- **Cancel** button: Cancels the Print Wizard, without creating any printout.

Additional Content

This page appears as part of the **Advanced Print Wizard** and the Output Creation Preset Wizard.

The **Additional Content** page allows you to add content at the time of printing. There are four different types of Additional Content that can be added at print time: **Text**; **Images**; **Barcodes** and **OMR Marks**.
They can be used to add either static content or variable content.

The Additional Content option is particularly useful when you might need to drive custom processes on production machines using either Barcodes or OMR Marks, or if you need to add some last minute additions to the print job via text and/or images.

Page breakdown

The Additional Content table displays any Additional Content that has been set.
Additional Content can be added, edited, removed, duplicated or moved within the table via the buttons to the right of the table.

The **Additional Content options** are:

- **Add**: Click to open some Additional Content.
  The choices are as follows:
  - "Additional Text Settings" on page 734 entry.
  - "Additional Barcode Options" on page 737 entry, from the selection presented.
  - "Additional Image Settings" on page 735 entry.
  - "Additional OMR Mark Settings" on page 768 entry.
- **Edit**: Click to edit the currently selected entry. This will launch the appropriate edit box, for the selected entry type.
  Double clicking on an entry in the table has the same effect as this button.
- **Delete**: Click to delete the currently selected entry or entries.
- **Duplicate**: Click to create a duplicated copy of the entry. This creates a new entry that is exactly the same as the original.
- / **Move Up / Move Down**: Click to move the selected entry (or entries) up or down within the table.

Additional Content Table: The table is split up into the following columns

- **Include in Output**: Indicates whether the entry is to be included in the output or not.
  The settings made in the Additional Content dialog box can be over-ridden here.
  This allows you to create a virtual "library" of Additional Content options in an Output Preset, from which you can then pick and choose from at time of printing.
- **Type**: Displays an icon showing what type the Additional Content entry is.
- **Description**: Displays the Description" that was entered in the Additional Content dialog.

> **Note**
>
> The entered text Description might stretch over a couple of lines, so it is possible that not all of the Description will be displayed here.

- **Left**: Displays the Additional Content Left Positional value.
- **Bottom**: Displays the Additional Content Bottom Positional value.
- **Content**: Displays the actual Additional Content that is to be inserted.
  In the case of Text, this would be the actual text (or data fields) that are to be inserted.
  In the case of Image, this would be the name of the image file.
  In the case of Barcode, this would be the data fields used for creating the barcode.
  In the case of OMR, this would be the number of marks set for this entry.
- **Condition**: Displays the Conditional entry that is used to determine whether this Additional Content instance is to be added to the output or not.

Advanced Print Wizard and Output Creation Preset Wizard navigation options

- **Preview** button (active in ***Advanced Print*** Wizard only): Click to launch a Proof Preview window, which displays how the printed output would look based upon the currently chosen selections.
- **Back** and **Next** buttons: Used to navigate back and forth through all the selected options within the Wizard. Up until the Print button is pressed, one can reverse all the way through the wizard to return to the main selection page (the "Print options" on page 704 page) and add or remove printing options from the print run.
- **Print** button (***Advanced Print*** Wizard only) or **Finish** button (***Output Creation Preset*** Wizard only):
  Click to produce print output/finalize the Preset according to the current settings.
  This can be done at any point within the Wizard, whether or not *all* the options selected in

the "Print options" on page 704 page have been completed or not.

- **Cancel** button: Cancels the Print Wizard, without creating any printout.

Additional Text Settings

The **Additional Text Settings** dialog displays the property of Text added in the "Additional Content" on page 732 page.

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.
  - **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.
- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the Text added to the page.
  - **Output once per sheet**: Option relates to Imposition (also known as N-Up) printing. Select this box to have the Text printed once per sheet rather than once per document page.

    > **Note**
    >
    > If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

  - **Left**: Enter the distance between the left margin of the page and the Text, in either metric (cm/mm), inch (in), pixel (px) or point (pt) values.
  - **Bottom**: Enter the distance between the bottom margin of the page and the Text, in either metric (cm/mm), inch (in), pixel (px) or point (pt) values.
- **Font** group:

- **Font Name**: Use the drop-down to select which font type to apply to the Text. The drop-down displays all the fonts installed on the system.

  > **Note**
  >
  > Vertically-oriented fonts (fonts whose typeface name begin with the at (@) character) are not supported in Connect.

  - **Style**: Choose between "Regular", "*Italic*", "**Bold**" or "***Bold Italic***".
  - **Font Size**: Enter the font size in points (pt).
  - **Color**: Select what color the Text will be.
- **Text**: Enter the actual Text to appear on the page in the selected location. The Text can be spread over multiple lines, but no additional formatting can be added within this edit box. The entire Text will be printed using the formatting options selected in the **Font group**.

  Use the **Add** button to display a list of variable data that can be added to the Text.

  This includes metadata fields added in the Metadata Options, as well as some document information fields (see "Print output variables" on page 1008 for information on those).
- **Condition**: Enter the condition which determines whether or not the Text will be added to the document at print time.

  Use the **Add** button for selection options.

  For details on how to create a conditional, see the "How to Set Conditionals" on page 772 page.

Additional Image Settings

The **Additional Image** dialog displays the properties of the image added in the "Additional Content" on page 732 page.

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets,

libraries from which you can pick and choose what entries you wish to have included, at time of printing.

- **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.

- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the image.
  - **Layer**: Whether this image will appear behind the text (the text will print over the image) or in front of the text (the text behind will be blanked out by the image, as transparent images are not supported).
    - **Ouput once per sheet**: Option relates to Imposition (also known as N-Up) printing. Select this box to have the Image printed once per sheet rather than once per document page.

> **Note**
>
> If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

  - **Left**: Enter the distance between the left margin of the page and the image, in either metric (cm/mm), inch (in), pixel (px) or point (pt) values.
  - **Bottom**: Enter the distance between the bottom margin of the page and the image, in either metric (cm/mm), inch (in), pixel (px) or point (pt) values.
- **Filename**: Use the browse button to launch a Browse box to select an image file. This is a mandatory field.

> **Note**
>
> Transparent images are not supported.

- **Preview** group: This displays a preview of the selected Image.
- **Scaling** group:
  Scaling the image expands the image but keeps the aspect ratio. The amount of scale

and specific limitations can be applied used a combination of the following options:

- **Max Width**: Enter the absolute maximum width the image can be scaled to, in either metric (cm/mm), inch (in), pixel (px) or point (pt) values.
- **Max Height**: Enter the absolute maximum height the image can be scaled to, in either metric (cm/mm), inch (in), pixel (px) or point (pt) values.
- **Scale**: What scale to apply to the image. The maximum scale is 10.0 to 1. Decimal values are allowed for this field.

- **Condition**: Enter the condition which determines whether or not the image will be added to the document at print time. Use the button for selection options.

  For details on how to create a conditional, see "How to Set Conditionals" on page 772 page.

Additional Barcode Options

When adding Barcodes in the "Additional Content" on page 732 page you can select from a series of predetermined Barcode types.
To create dynamic barcodes, "Meta Data options" on page 699 must first have been set. Metadata fields are required to create the association between the dynamic data used in the print run and the barcode.

The options for each of these types are described on the following pages:

- "Australia Post Settings" on the next page
- "Codabar Settings" on page 738
- "Code 39 Settings" on page 743
- "Code 128 Settings" on page 740
- "Data Matrix settings" on page 745
- "EAN-8 Settings" on page 753
- "EAN-13 Settings" on page 751
- "GS1-128 Settings" on page 748
- "Interleaved 2 of 5 Settings" on page 755
- "Japan Post Settings" on page 652
- "KIX Code (Dutch Post) Settings" on page 654

- "PDF417 Settings" on page 757
- "QR Code Settings" on page 760
- "Royal Mail 2D Mailmark Settings" on page 662
- "Royal Mail 4 State (CBC) Settings" on page 664
- "Royal Mail 4 State Mailmark C Settings" on page 666
- "Royal Mail 4 State Mailmark L Settings" on page 668
- "UPC-A Settings" on page 763
- "UPC-E Settings" on page 766
- "US Postal Service IMb Settings" on page 675
- "US Postal Service IMpb Settings" on page 677

Australia Post Settings

Australia Post 4 State barcodes are specific to Australian postal addresses. Every address in Australia has a numeric Delivery Point IDs (DPID) associated with it. Addresses must first be matched against an official Postal Address File (PAF), to obtain the unique DPID, and the DPIDs must be available to the job as "Meta Data options" on page 699.

Use the following options to configure the output Barcode settings:

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.
  - **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.
- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the Barcode added to the page.
  - **Output once per sheet**: This option relates to Imposition printing (see Imposition), also known as N-Up printing. Select this box to have the Barcode printed once per sheet rather than once per document page.

> **Note**
>
> If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

  - **Left**: Enter the distance between the left margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
  - **Bottom**: Enter the distance between the bottom margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.

- **Australia Post 4 State Properties** group:
  - **Track height**: The height of the small vertical centre bars, in metric (mm/cm). Value must be between 1 mm and 1.6 mm.
  - **Full bar height**: Height of the tallest vertical bars, in metric (mm/cm). Value must be between 4.2 mm and 5.8 mm.

    The ascender and descender bar heights are automatically calculated based upon the *Full* and *Track* bar heights.
  - **Module width**: Thickness of each bar in the Barcode, in metric (mm/cm). Value must be between 0.4 mm and 0.6 mm.
  - **Module spacing**: The spacing between each bar, in metric (mm/cm). Value must be between 0.4 mm and 0.7 mm

- **Process Tilde**:
  This selection is not applicable to this barcode type.

- **Text**: Enter the text used to generate the Barcode.
  - **Add** button: Click to display a list of variable data that could be used for generating the Barcode.
    This includes metadata fields which must previously have been added in the Metadata options, likely at the *Document Tags* level (see Metadata Options), as well as some information fields.
- **Condition**: Enter the condition which determines whether or not the Barcode will be added to the document at print time. For details on how to create a conditional, see "How to Set Conditionals" on page 772.

- **Add** button: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

Codabar Settings

Codabar barcodes support the following data: 0-9 - $: / . + plus the optional specification of start/stop characters.

> **Note**
>
> To create dynamic barcodes, Metadata options must first have been set (see "Meta Data options" on page 699).
> Metadata fields are required to create the association between the dynamic data used in the print run and the barcode.

Use the following options to configure the output Barcode settings:

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.
  - **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.
- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the Barcode added to the page.
  - **Output once per sheet**: This option relates to Imposition printing (see Imposition), also known as N-Up printing. Select this box to have the Barcode printed once per sheet rather than once per document page.

> **Note**
>
> If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

- **Left**: Enter the distance between the left margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
- **Bottom**: Enter the distance between the bottom margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.

- **Codabar Properties** group:
  - **Height**: Enter the Barcode height in either metric (cm/mm), inch (in) or point (pt) values.
  - **Module Width**: Specifies the width of the narrow bars. Changing this value to a higher value will generally make the Barcode bigger.
  - **Bar width ratio**: Set the Barcode bar width.
  - **Default start symbol**: Use the drop-down to select the optional Barcode start character, which defines the encoding mode.
  - **Default stop symbol**: Use the drop-down to select the Barcode stop character, which defines the encoding mode.
    - **Print human readable text:** Check to add a textual version of the Barcode data.
    - **Placement**: Use the drop-down to select whether to place the human readable text above or below the Barcode.
    - **Font name**: Use the drop-down to select the font with which to display the human readable text.

      > **Note**
      >
      > Vertically-oriented fonts (fonts whose typeface name begin with the at (@) character) are not supported in Connect.

    - **Font size**: Enter a font size for the human readable text.

- **Display start/stop symbols** checkbox: Adds the stop/start symbols to the Barcode text.

- **Process Tilde**:
  This selection is not applicable to this barcode type.

- **Text**: Enter the text used to generate the Barcode.
  - **Add** button: Click to display a list of variable data that could be used for generating the Barcode.
    This includes metadata fields which must previously have been added in the Metadata options, likely at the *Document Tags* level (see Metadata Options), as well as some information fields.
- **Condition**: Enter the condition which determines whether or not the Barcode will be added to the document at print time. For details on how to create a conditional, see "How to Set Conditionals" on page 772.
  - **Add** button: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

Code 39 Settings

Code 39 is a discrete, self-checking barcode that is also known as "Alpha39", "Code 3 of 9" (often abbreviated to "3 of 9"), "Code 3/9", "Type 39", "USS Code 39" and "USD-3".

Code 39 data should contain no more than 20 digits from within the following range: Numeric digits: (0-9), upper-case letters (A-Z), seven special characters (- . space $ / + %) and the start/stop asterisk (*) character.

If the Extended character set is chosen, then lower-case letters (a-z) and other special ASCII characters can also be included.

> **Note**
>
> To create dynamic barcodes, Metadata options must first have been set (see "Meta Data options" on page 699).
> Metadata fields are required to create the association between the dynamic data used in the print run and the barcode.

Use the following options to configure the output Barcode settings:

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.
  - **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.
- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the Barcode added to the page.
  - **Output once per sheet**: This option relates to Imposition printing (see Imposition), also known as N-Up printing. Select this box to have the Barcode printed once per sheet rather than once per document page.

    > **Note**
    >
    > If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

  - **Left**: Enter the distance between the left margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
  - **Bottom**: Enter the distance between the bottom margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
- **Code 39 Properties** group:
  - **Height**: Enter the Barcode height in either metric (cm/mm), inch (in) or point (pt) values.
  - **Use extended character set**: Check to use the Code 39 Extended character set. This extends the range of supported data to include the full ASCII character set. This adds support for lower case letters (a-z) and the full range of ASCII punctuation and special characters.

- **Module Width**: Specifies the width of the narrow bars. Changing this value to a higher value will generally make the Barcode bigger. The smallest Module Width is 0.19mm (high density).
- **Bar width ratio**: Set the Barcode bar width.
    - **Checksum**: Use the drop-down to select how to deal with the Barcode checksum:
    - **Ignore**: Ignore checksum calculations.
    - **Auto**: Add a checksum character to the Barcode if the initial value does not validate. This is the default value.
    - **Check**: Verify the Barcode has a valid checksum.
    - **Add**: Calculate and add a checksum character to Barcode, regardless of current value.
    - **Print human readable text:** Check to add a textual version of the Barcode data.
    - **Placement**: Use the drop-down to select whether to place the human readable text above or below the Barcode.
    - **Font name**: Use the drop-down to select the font with which to display the human readable text.

      > **Note**
      >
      > Vertically-oriented fonts (fonts whose typeface name begin with the at (@) character) are not supported in Connect.

    - **Font size**: Enter a font size for the human readable text.
- **Process Tilde**:
  Check this option to process tilde characters in the data as special characters.

  Note: with this option checked, any tilde that needs to be included in the output must be escaped by adding another tilde: ~~.

- **Text**: Enter the text used to generate the Barcode.
    - **Add** button: Click to display a list of variable data that could be used for generating the Barcode.
      This includes metadata fields which must previously have been added in the

Metadata options, likely at the *Document Tags* level (see Metadata Options), as well as some information fields.

- **Condition**: Enter the condition which determines whether or not the Barcode will be added to the document at print time. For details on how to create a conditional, see "How to Set Conditionals" on page 772.
  - **Add** button: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

Code 128 Settings

Code 128 is a high-density barcode, used for alphanumeric or numeric-only barcodes. It supports all 128 ASCII characters.

> **Note**
>
> To create dynamic barcodes, Metadata options must first have been set (see "Meta Data options" on page 699).
> Metadata fields are required to create the association between the dynamic data used in the print run and the barcode.

Use the following options to configure the output Barcode settings:

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.
  - **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.
- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the Barcode added to the page.
  - **Output once per sheet**: This option relates to Imposition printing (see Imposition), also known as N-Up printing. Select this box to have the Barcode printed once per sheet rather than once per document page.

> **Note**
>
> If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

  - **Left**: Enter the distance between the left margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
  - **Bottom**: Enter the distance between the bottom margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.

- **Code 128 Properties** group:
- **Height**: Enter the Barcode height in either metric (cm/mm), inch (in) or point (pt) values.
- **Module Width**: Specifies the width of the narrow bars. Changing this value to a higher value will generally make the Barcode bigger.
  - **Print human readable text:** Check to add a textual version of the Barcode data.
  - **Placement**: Use the drop-down to select whether to place the human readable text above or below the Barcode.
  - **Font name**: Use the drop-down to select the font with which to display the human readable text.

> **Note**
>
> Vertically-oriented fonts (fonts whose typeface name begin with the at (@) character) are not supported in Connect.

  - **Font size**: Enter a font size for the human readable text.

- **Process Tilde**:
  Check this option to process tilde characters in the data as special characters.

  Note: with this option checked, any tilde that needs to be included in the output must be escaped by adding another tilde: ~~.

- **Text**: Enter the text used to generate the Barcode.
  - **Add** button: Click to display a list of variable data that could be used for generating the Barcode.
    This includes metadata fields which must previously have been added in the Metadata options, likely at the *Document Tags* level (see Metadata Options), as well as some information fields.
- **Condition**: Enter the condition which determines whether or not the Barcode will be added to the document at print time. For details on how to create a conditional, see "How to Set Conditionals" on page 772.
  - **Add** button: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

Data Matrix settings

A Data Matrix barcode is a high-density, two-dimensional (2D) matrix barcode which supports encoded text, numbers, files and digital data.

Note

In order to create a **GS1** compliant Data Matrix barcode, the barcode data must meet the following requirements:

- The barcode data must start with a leading FNC1 character. Either ~1 or ~d232.
- The GS1 Application Identifiers (AI) must be used for all data.
  The function code ~1 must be used as field separator for variable length AI elements.
- Only ASCII characters should be used.

Use the following options to configure the output Barcode settings:

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have

included, at time of printing.

- **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.

- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the Barcode added to the page.
  - **Output once per sheet**: This option relates to Imposition printing (see Imposition), also known as N-Up printing. Select this box to have the Barcode printed once per sheet rather than once per document page.

    > **Note**
    >
    > If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

  - **Left**: Enter the distance between the left margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
  - **Bottom**: Enter the distance between the bottom margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.

- **Data Matrix Properties** group:
  - **Module Width**: Specifies the width of the narrow bars. Changing this value to a higher value will generally make the Barcode bigger.
  - **Encoding**: The data represented in the symbol can be compressed using one of the following algorithms:
    - **Auto**: Automatically detect the data content and encodes using the most appropriate method. This is the default option.
    - **ASCII**: is used to encode data that mainly contains ASCII alphanumeric characters (ASCII 0-127). Use where Barcode size is a concern and where the data is alphanumeric.
    - **Base 256**: used to encode 8-bit values.
    - **C40**: used for data that mainly consists of numbers and upper-case alphabetic letters.

- **Text**: used for data that mainly consists of numbers and lower-case alphabetic letters.
- **None**: Does not use any encoding.

- **Format**: select the Barcode size format from the drop-down list .

- **Process Tilde**:
  Check this option to process tilde characters in the data as special characters.

  Note: with this option checked, any tilde that needs to be included in the output must be escaped by adding another tilde: ~~.

  Note

  The following tilde codes are supported in GS1 DataMatrix barcodes:

  - ~1 = FNC1 character (for GS1 DataMatrix barcodes)
  - ~2 = Structure Append
  - ~3 = Reader programming
  - ~5 = Macro5
  - ~6 = Macro6
  - ~7 = ECI expressions

- **Text**: Enter the text used to generate the Barcode.
  - **Add** button: Click to display a list of variable data that could be used for generating the Barcode.
    This includes metadata fields which must previously have been added in the Metadata options, likely at the *Document Tags* level (see Metadata Options), as well as some information fields.
- **Condition**: Enter the condition which determines whether or not the Barcode will be added to the document at print time. For details on how to create a conditional, see "How to Set Conditionals" on page 772.
  - **Add** button: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

EAN-8 Settings

An EAN-8 barcodes are composed entirely of numerical data. It is comprised of 7 data digits containing the country/economic area code and an item reference code, with 1 following checksum digit.

Use the following options to configure the output Barcode settings:

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.
  - **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.
- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the Barcode added to the page.
  - **Output once per sheet**: This option relates to Imposition printing (see Imposition), also known as N-Up printing. Select this box to have the Barcode printed once per sheet rather than once per document page.

    > **Note**
    >
    > If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

  - **Left**: Enter the distance between the left margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
  - **Bottom**: Enter the distance between the bottom margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.

- **EAN-8 Properties**:
  - **Height**: Enter the Barcode height in either metric (cm/mm), inch (in) or point (pt) values.
  - **Module Width**: Specifies the width of the narrow bars. Changing this value to a higher value will generally make the Barcode bigger. The EAN-8 barcode employs a module width between 0.27mm and 0.66mm.
  -
    - **Checksum**: Use the drop-down to select how to deal with the Barcode checksum:
    - **Ignore**: Ignore checksum calculations.
    - **Auto**: Add a checksum character to the Barcode if the initial value does not validate. This is the default value.
    - **Check**: Verify the Barcode has a valid checksum.
    - **Add**: Calculate and add a checksum character to Barcode, regardless of current value.
    - **Print human readable text:** Check to add a textual version of the Barcode data.
    - **Placement**: Use the drop-down to select whether to place the human readable text above or below the Barcode.
    - **Font name**: Use the drop-down to select the font with which to display the human readable text.

      > **Note**
      >
      > Vertically-oriented fonts (fonts whose typeface name begin with the at (@) character) are not supported in Connect.

    - **Font size**: Enter a font size for the human readable text.
- **Process Tilde**:
  This selection is not applicable to this barcode type.
- **Text**: Enter the text used to generate the Barcode.
  - **Add** button: Click to display a list of variable data that could be used for generating the Barcode.
    This includes metadata fields which must previously have been added in the

Metadata options, likely at the *Document Tags* level (see Metadata Options), as well as some information fields.

- **Condition**: Enter the condition which determines whether or not the Barcode will be added to the document at print time. For details on how to create a conditional, see "How to Set Conditionals" on page 772.
  - **Add** button: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

EAN-13 Settings

EAN-13 barcodes are composed entirely of numerical data. The first 12 digits representing country/economic area, manufacturer and product codes + 1 following checksum digit.

> **Note**
>
> To create dynamic barcodes, Metadata options must first have been set (see "Meta Data options" on page 699).
> Metadata fields are required to create the association between the dynamic data used in the print run and the barcode.

Use the following options to configure the output Barcode settings:

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.
  - **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.
- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the Barcode added to the page.
  - **Output once per sheet**: This option relates to Imposition printing (see Imposition), also known as N-Up printing. Select this box to have the Barcode printed once per sheet rather than once per document page.

> **Note**
>
> If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

  - **Left**: Enter the distance between the left margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
  - **Bottom**: Enter the distance between the bottom margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.

- **EAN 13 Properties** group:
  - **Height**: Enter the Barcode height in either metric (cm/mm), inch (in) or point (pt) values.
  - **Module Width**: Specifies the width of the narrow bars. Changing this value to a higher value will generally make the Barcode bigger. The EAN-13 barcode employs a module width between 0.27mm and 0.66mm.
    - **Checksum**: Use the drop-down to select how to deal with the Barcode checksum:
    - **Ignore**: Ignore checksum calculations.
    - **Auto**: Add a checksum character to the Barcode if the initial value does not validate. This is the default value.
    - **Check**: Verify the Barcode has a valid checksum.
    - **Add**: Calculate and add a checksum character to Barcode, regardless of current value.
    - **Print human readable text:** Check to add a textual version of the Barcode data.
    - **Placement**: Use the drop-down to select whether to place the human readable text above or below the Barcode.
    - **Font name**: Use the drop-down to select the font with which to display the human readable text.

> **Note**
>
> Vertically-oriented fonts (fonts whose typeface name begin with the at (@) character) are not supported in Connect.

  - **Font size**: Enter a font size for the human readable text.

- **Process Tilde**:
  This selection is not applicable to this barcode type.

- **Text**: Enter the text used to generate the Barcode.
  - **Add** button: Click to display a list of variable data that could be used for generating the Barcode.
    This includes metadata fields which must previously have been added in the Metadata options, likely at the *Document Tags* level (see Metadata Options), as well as some information fields.
- **Condition**: Enter the condition which determines whether or not the Barcode will be added to the document at print time. For details on how to create a conditional, see "How to Set Conditionals" on page 772.
  - **Add** button: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

GS1-128 Settings

GS1-128 is also known as "EAN 128", "EAN/UCC 128" and "UCC 128". This barcode type not only encodes data, but also provides a mechanism for defining the meaning (or format) of that data. It supports alphanumeric data and some predefined Function Codes. See the Wikipedia GS1-128 entry for more information.

> **Note**
>
> To create dynamic barcodes, Metadata options must first have been set (see "Meta Data options" on page 699).
> Metadata fields are required to create the association between the dynamic data used in

the print run and the barcode.

Use the following options to configure the output Barcode settings:

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.
  - **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.
- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the Barcode added to the page.
  - **Output once per sheet**: This option relates to Imposition printing (see Imposition), also known as N-Up printing. Select this box to have the Barcode printed once per sheet rather than once per document page.

    Note

    If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

  - **Left**: Enter the distance between the left margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
  - **Bottom**: Enter the distance between the bottom margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
- **GS1-128 Properties** group:
  - **Height**: Enter the Barcode height in either metric (cm/mm), inch (in) or point (pt) values.

- **Module Width**: Specifies the width of the narrow bars. Changing this value to a higher value will generally make the Barcode bigger.
- **Check Digit marker**: This character is used as a placeholder for the check digit, which we be calculated at runtime. The character must be expressed in Hex.
- **Group separator**: This character is used to define group separation points. The character must be expressed in Hex.
- **Template**: Specify an optional Barcode "template".
  Examples:
  - `n13` defines a numeric field with exactly 13 digits.
  - `n13+cd` defines a numeric field with exactly 13 digits plus a check digit.
  - `an1-9` defines an alpha-numeric field with 1 to 9 characters.

  Elements can be combined using the '+' symbol.
  - **Print human readable text:** Check to add a textual version of the Barcode data.
  - **Placement**: Use the drop-down to select whether to place the human readable text above or below the Barcode.
  - **Font name**: Use the drop-down to select the font with which to display the human readable text.

    > **Note**
    >
    > Vertically-oriented fonts (fonts whose typeface name begin with the at (@) character) are not supported in Connect.

  - **Font size**: Enter a font size for the human readable text.

- **Process Tilde**:
  Check this option to process tilde characters in the data as special characters.

  Note: with this option checked, any tilde that needs to be included in the output must be escaped by adding another tilde: ~~.

- **Text**: Enter the text used to generate the Barcode.
  - **Add** button: Click to display a list of variable data that could be used for generating the Barcode.
    This includes metadata fields which must previously have been added in the Metadata options, likely at the *Document Tags* level (see Metadata Options), as well as some information fields.
- **Condition**: Enter the condition which determines whether or not the Barcode will be added to the document at print time. For details on how to create a conditional, see "How to Set Conditionals" on page 772.
  - **Add** button: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

Interleaved 2 of 5 Settings

Interleaved 2 of 5 barcodes are also known as "ITF" and "2/5 Interleaved". It is a numeric only barcode whose data must contain an even number of digits, as the barcode uses sequences of two digits interleaved with each other to create a single symbol. If the numeric data contains an odd number of digits, then a leading zero must be added to the beginning of the data.

Use the following options to configure the output Barcode settings:

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.
  - **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.
- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the Barcode added to the page.
  - **Output once per sheet**: This option relates to Imposition printing (see Imposition), also known as N-Up printing. Select this box to have the Barcode printed once per sheet rather than once per document page.

> **Note**
>
> If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

  - **Left**: Enter the distance between the left margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
  - **Bottom**: Enter the distance between the bottom margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.

- **Interleaved 2 of 5 Properties** group:
  - **Height**: Enter the Barcode height in either metric (cm/mm), inch (in) or point (pt) values.
  - **Module Width**: Specifies the width of the narrow bars. Changing this value to a higher value will generally make the Barcode bigger.
  - **Bar width ratio**: Set the Barcode bar width.
    - **Print human readable text:** Check to add a textual version of the Barcode data.
    - **Placement**: Use the drop-down to select whether to place the human readable text above or below the Barcode.
    - **Font name**: Use the drop-down to select the font with which to display the human readable text.

      > **Note**
      >
      > Vertically-oriented fonts (fonts whose typeface name begin with the at (@) character) are not supported in Connect.

    - **Font size**: Enter a font size for the human readable text.

- **Process Tilde**:
  This selection is not applicable to this barcode type.

- **Text**: Enter the text used to generate the Barcode.
  - **Add** button: Click to display a list of variable data that could be used for generating the Barcode.
    This includes metadata fields which must previously have been added in the Metadata options, likely at the *Document Tags* level (see Metadata Options), as well as some information fields.
- **Condition**: Enter the condition which determines whether or not the Barcode will be added to the document at print time. For details on how to create a conditional, see "How to Set Conditionals" on page 772.
  - **Add** button: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

Japan Post Settings

Japan Post 4 State Barcodes (also known as Kasutama barcodes, or Japanese Postal Barcode) are used by Japan Post for automated mail sorting. These barcodes contain the address of the receiver encoded as a seven-digit postal code plus optional address data of up to 13 alphanumeric characters.

> **Note**
>
> To create dynamic barcodes, Metadata options must first have been set (see "Meta Data options" on page 699).
> Metadata fields are required to create the association between the dynamic data used in the print run and the barcode.

Use the following options to configure the output Barcode settings:

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.

- **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.

- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the Barcode added to the page.
  - **Output once per sheet**: This option relates to Imposition printing (see Imposition), also known as N-Up printing. Select this box to have the Barcode printed once per sheet rather than once per document page.

    > **Note**
    >
    > If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

  - **Left**: Enter the distance between the left margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
  - **Bottom**: Enter the distance between the bottom margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.

- **Japan Post Properties** group:
  - **Track height**: The height of the small vertical centre bars, in metric (mm/cm). Value must be between 0.6 mm and 1.8 mm.
  - **Full bar height**: Height of the tallest vertical bars, in metric (mm/cm). Value must be between 1.8 mm and 5.4 mm.
  - **Module width**: Thickness of each bar in the Barcode, in metric (mm/cm). Value must be between 0.3 mm and 0.9 mm.
  - **Module spacing**: The spacing between each bar, in metric (mm/cm). Value must be between 0.3 mm and 0.9 mm
- **Process Tilde**:
  This selection is not applicable to this barcode type.

- **Text**: Enter the text used to generate the Barcode.
  - **Add** button: Click to display a list of variable data that could be used for generating the Barcode.
    This includes metadata fields which must previously have been added in the

Metadata options, likely at the *Document Tags* level (see Metadata Options), as well as some information fields.

- **Condition**: Enter the condition which determines whether or not the Barcode will be added to the document at print time. For details on how to create a conditional, see "How to Set Conditionals" on page 772.
  - **Add** button: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

KIX Code (Dutch Post) Settings

KIX code barcodes (also known as Klantenindex, or Dutch KIX 4-State), are used by Royal Dutch TPG Post (Netherlands) for Postal code and automatic mail sorting. They contain the address of the receiver, encoded in alphanumeric format (0-9, A-Z).

> **Note**
>
> To create dynamic barcodes, Metadata options must first have been set (see "Meta Data options" on page 699).
> Metadata fields are required to create the association between the dynamic data used in the print run and the barcode.

Use the following options to configure the output Barcode settings:

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.
  - **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.
- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the Barcode added to the page.
  - **Output once per sheet**: This option relates to Imposition printing (see Imposition), also known as N-Up printing. Select this box to have the Barcode printed once per

sheet rather than once per document page.

> **Note**
>
> If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

- **Left**: Enter the distance between the left margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
- **Bottom**: Enter the distance between the bottom margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.

- **KIX Code Properties** group:
  - **Track height**: The height of the small vertical centre bars, in metric (mm/cm). Value must be between 1 mm and 1.6 mm.
  - **Full bar height**: Height of the tallest vertical bars, in metric (mm/cm). Value must be between 4.2 mm and 5.8 mm.

    The ascender and descender bar heights are automatically calculated based upon the *Full* and *Track* bar heights.
  - **Module width**: Thickness of each bar in the Barcode, in metric (mm/cm). Value must be between 0.4 mm and 0.6 mm.
  - **Module spacing**: The spacing between each bar, in metric (mm/cm). Value must be between 0.4 mm and 0.7 mm

- **Process Tilde**:
  This selection is not applicable to this barcode type.

- **Text**: Enter the text used to generate the Barcode.
  - **Add** button: Click to display a list of variable data that could be used for generating the Barcode.
    This includes metadata fields which must previously have been added in the Metadata options, likely at the *Document Tags* level (see Metadata Options), as well as some information fields.

- **Condition**: Enter the condition which determines whether or not the Barcode will be added to the document at print time. For details on how to create a conditional, see "How to Set Conditionals" on page 772.
    - **Add** button: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

PDF417 Settings

PDF417 is a two-dimensional, multi-row Barcode. It is used for encoding large amounts of data, with hundreds or even thousands of characters. It encodes alphabetic text, numbers, binary files and actual data bytes.

> **Note**
>
> To create dynamic barcodes, Metadata options must first have been set (see "Meta Data options" on page 699).
> Metadata fields are required to create the association between the dynamic data used in the print run and the barcode.

Use the following options to configure the output Barcode settings:

- **General** group:
    - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
      It is checked by default for all new entries.
      This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.
    - **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.
- **Position** group:
    - **Orientation**: Use the drop-down to select the orientation of the Barcode added to the page.
    - **Output once per sheet**: This option relates to Imposition printing (see Imposition), also known as N-Up printing. Select this box to have the Barcode printed once per sheet rather than once per document page.

> **Note**
>
> If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

  - **Left**: Enter the distance between the left margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
  - **Bottom**: Enter the distance between the bottom margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.

- **PDF417 Properties** group:
  - **Module Width**: Specifies the width of the narrow bars. Changing this value to a higher value will generally make the Barcode bigger.
  - **Row height**: Defines the height of the bars for a single row, measured in pixels, points or metric.
  - **Width to height ratio**: Select the ratio of column width to row height.
  - **Mode**: Use the drop-down to set the compaction mode.
    - **Binary**: allows any byte value to be encoded.
    - **Text**: allows all printable ASCII characters to be end coded (ASCII values 32 to 126 and some additional control characters).
    - **Numeric**: more efficient mode for encoding numeric data.
    - **Auto**: Automatically detect the data content and encodes using the most appropriate method. This is the default option.
  - **Error Correction Level**: Enter the error correction level for the built-in error correction method based on Reed-Solomon algorithms. The error correction level is adjustable between level 0 (just error detection, without correction) and level 8 (maximum error correction). Recommended error correction levels are between level 2 and 5, but the optimal value depends on the amount of data, printing quality of the PDF417 symbol and decoding capabilities.
  - **Rows**: A PDF417 bar code can have anywhere from 3 to 90 rows.
  - **Columns**: The number of data columns can vary from 1 to 30.

- **Tilde processing**

  Check this option to process tilde (~) characters in the data as special characters. (See the Java4less Barcodes Guide to learn what the tilde character can be used for.)
  The tilde is expected to be followed by the 'd' character and 3 digits representing an ASCII character: **~dNNN** . For example, ~d013 represents a carriage return.
  Note that with this option checked, any tilde that needs to be included in the output must be escaped by adding another tilde: ~~.

- **Text**: Enter the text used to generate the Barcode.
  - **Add** button: Click to display a list of variable data that could be used for generating the Barcode.
    This includes metadata fields which must previously have been added in the Metadata options, likely at the *Document Tags* level (see Metadata Options), as well as some information fields.
- **Condition**: Enter the condition which determines whether or not the Barcode will be added to the document at print time. For details on how to create a conditional, see "How to Set Conditionals" on page 772.
  - **Add** button: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

QR Code Settings

QR Code (Quick Response Code) is a 2D Barcode format that supports alphanumeric, numeric, byte/binary, and Kanji (Japanese-Chinese character) data.

> **Note**
>
> To create dynamic barcodes, Metadata options must first have been set (see "Meta Data options" on page 699).
> Metadata fields are required to create the association between the dynamic data used in the print run and the barcode.

Use the following options to configure the output Barcode settings:

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.
  - **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.
- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the Barcode added to the page.
  - **Output once per sheet**: This option relates to Imposition printing (see Imposition), also known as N-Up printing. Select this box to have the Barcode printed once per sheet rather than once per document page.

    > **Note**
    >
    > If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

  - **Left**: Enter the distance between the left margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
  - **Bottom**: Enter the distance between the bottom margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.

QRCode Properties group:

- **Size by**: Select size from the two options available:
  - **By area**: Connect will try to size the Barcode to fit the specified area by dynamically changing the module width to the **Size** selection. The lower module width limit is governed by the **Minimum module width** selection.
    Enter the sizes in either metric (cm/mm), inch (in), pixel (px) or point (pt) values.
  - **By module width**: Connect will try to size the Barcode to the module width of the characters. Large Barcode values will result in larger Barcode and vice versa.

Enter the **Module width**in either metric (cm/mm), inch (in), pixel (px) or point (pt) values.

- **Encoding**: Define the encoding of the Barcode:
  - **Auto**: Automatically detect the data content and encodes using the most appropriate method. This is the default option.
  - **Numeric**: 7089 numerical characters.
  - **Alphanumeric**: 4296 alphanumerical characters.
  - **Byte**: 2953 characters.
  - **Kanji**: 1817 Japanese/Chinese characters.
- **Version**: Select the preferred QR code version (which sets the data length field) from the 40 available.

> **Note**
>
> The Encoding and Version fields work together to determine how many characters are encoded within a *length field*. The following table shows the number of bits in a length field, based upon the selections made:

Encoding	Ver. 1-9	Ver. 10-23	Ver. 27-40
Numeric	10	12	14
Alphanumeric	9	11	13
Byte	8	16	16
Kanji	8	10	12

- **Error Correction Level**: Part of the robustness of QR codes is their ability to sustain “damage” and continue to function even when a part of the QR code image is obscured, defaced or removed. A higher correction level duplicates data within the QR Code to allow for damaged areas. The higher the Error Correction Level, the larger the Barcode will be. The choices are (in order from lowest to highest): **L**ow, **M**edium, **Q**uartile and **H**igh.
- **Use ECI for encoding messages as bytes**: Selecting Extended Channel Interpretations (ECI) allows encoding multiple character sets (e.g. Arabic, Cyrillic, Greek, Hebrew) and other data interpretations, into one QR Code symbol.

- **Multi-part QR Code (structured append)**: Select to append a QR Code symbol in a structured format.
  - **Part**: indicates the position of the QR Code symbol within the group of Structured Append symbols.
  - **of**: indicates how many Structured Append symbols exist.

> **Note**
>
> The Structured Append symbols Part number can never exceed the sum total of Structured Append symbols available (the "of" value). Thus selecting a Part number beyond the existing sum total will increase the sum total to the same value.

- **Use FNC1**: Check to enable Application Identifiers. These are often used to encode links to websites, or to encode production/batch details.
  - **Position**: Select between the two methods for encoding FNC1 characters within QR Codes:
    - **First** Position - uses the GS1 QR Code standard.
    - **Second** Position - uses the AIM QR Code standard. If this option is chosen then the appropriate Application Indicator will also need to be set.
      - **Application ID**: Enter the appropriate QR-Code Application Indicator in accordance with the specific industry or application specifications (as provided by AIM International).

- **Process Tilde**:
  Check this option to process tilde characters in the data as special characters.

  Note: with this option checked, any tilde that needs to be included in the output must be escaped by adding another tilde: ~~.

- **Text**: Enter the text used to generate the Barcode.
  - **Add** button: Click to display a list of variable data that could be used for generating the Barcode.
    This includes metadata fields which must previously have been added in the Metadata options, likely at the *Document Tags* level (see Metadata Options), as well as some information fields.

- **Condition**: Enter the condition which determines whether or not the Barcode will be added to the document at print time. For details on how to create a conditional, see "How to Set Conditionals" on page 772.
    - **Add** button: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

Royal Mail 2D Mailmark Settings

Royal Mail 2D Mailmark Barcodes are a variant of 2 dimensional Datamatrix barcodes. They are used by the British Royal Mail postal service. They contain the address of the receiver, encoded in alpha-numeric format (0-9, A-Z).

> **Note**
>
> To create dynamic barcodes, Metadata options must first have been set (see "Meta Data options" on page 699).
> Metadata fields are required to create the association between the dynamic data used in the print run and the barcode.

Use the following options to configure the output Barcode settings:

- **General** group:
    - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
      It is checked by default for all new entries.
      This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.
    - **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.
- **Position** group:
    - **Orientation**: Use the drop-down to select the orientation of the Barcode added to the page.
    - **Output once per sheet**: This option relates to Imposition printing (see Imposition), also known as N-Up printing. Select this box to have the Barcode printed once per sheet rather than once per document page.

> **Note**
>
> If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

  - **Left**: Enter the distance between the left margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
  - **Bottom**: Enter the distance between the bottom margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.

- **Royal Mail 2D Properties** group:
  - **Module Width**: The size of each column and row within the barcode. Value must be between 0.5 mm and 0.7 mm.
  - **Preferred Version**: Select the preferred size of the barcode in terms of rows by columns.

> **Note**
>
> Depending on the information being embedded, this entry may be overridden if the specified size is not big enough to contain the barcode information.

- **Tilde processing**

  Check this option to process tilde (~) characters in the data as special characters. (See the Java4less Barcodes Guide to learn what the tilde character can be used for.)
  The tilde is expected to be followed by the 'd' character and 3 digits representing an ASCII character: **~dNNN** . For example, ~d013 represents a carriage return.
  Note that with this option checked, any tilde that needs to be included in the output must be escaped by adding another tilde: ~~.

- **Text**: Enter the text used to generate the Barcode.
  - **Add** button: Click to display a list of variable data that could be used for generating the Barcode.
    This includes metadata fields which must previously have been added in the

Metadata options, likely at the *Document Tags* level (see Metadata Options), as well as some information fields.

- **Condition**: Enter the condition which determines whether or not the Barcode will be added to the document at print time. For details on how to create a conditional, see "How to Set Conditionals" on page 772.
  - **Add** button: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

Royal Mail 4 State (CBC) Settings

Royal Mail 4-State Customer Bar Code (CBC) is also known as the Royal Mail 4-State Customer Code (RM4SCC or RoyalMail4SCC). It is used by the British Royal Mail postal service. The barcode contains an address, encoded in alpha-numeric format (0-9, A-Z).

Use the following options to configure the output Barcode settings:

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.
  - **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.
- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the Barcode added to the page.
  - **Output once per sheet**: This option relates to Imposition printing (see Imposition), also known as N-Up printing. Select this box to have the Barcode printed once per sheet rather than once per document page.

> **Note**
>
> If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-

> Up sheet.

  - **Left**: Enter the distance between the left margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
  - **Bottom**: Enter the distance between the bottom margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.

- **Royal Mail Properties** group:
  - **Track height**: The height of the small vertical centre bars, in metric (mm/cm). Value must be between1.02 mm and 1.52 mm.
  - **Full bar height**: Height of the tallest vertical bars, in metric (mm/cm). Value must be between4.22 mm and 5.84 mm.

    The ascender and descender bar heights are automatically calculated based upon the *Full* and *Track* bar heights.
  - **Module width**: Thickness of each bar in the Barcode, in metric (mm/cm). Value must be between0.38 mm and 0.63 mm.
  - **Module spacing**: The spacing between each bar, in metric (mm/cm). Value must be between0.38 mm and 0.63 mm.

- **Process Tilde**:
  This selection is not applicable to this barcode type.

- **Text**: Enter the text used to generate the Barcode.
  - **Add** button: Click to display a list of variable data that could be used for generating the Barcode.
    This includes metadata fields which must previously have been added in the Metadata options, likely at the *Document Tags* level (see Metadata Options), as well as some information fields.
- **Condition**: Enter the condition which determines whether or not the Barcode will be added to the document at print time. For details on how to create a conditional, see "How to Set Conditionals" on page 772.
  - **Add** button: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

Royal Mail 4 State Mailmark C Settings

Royal Mail 4-State Mailmark C is used by the British Royal Mail postal service. The barcode contains an address, encoded in alpha-numeric format (0-9, A-Z).

The content (or message) of the barcode consists of the following 6 parts (or fields), which are detailed here:

Field	Length	Allowed Characters
Format	1	`[01234]`
Version ID	1	`[1234]`
Class	1	`[0123456789ABCDE]`
Supply Chain ID	2	`[0123456789]`
Item ID	8	`[0123456789]`
Destination Post Code plus DPS	9	`[0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ]` The value "XY11 " is used for this field to specify an *International Designation*.

As an example, the message `"1129966448822CC00AA0A "` has *Format* = 1, *Version ID* = 1, *Class* = 2, *Supply Chain ID* = 99, *Item ID* = 66448822 and *Destination Post Code plus DPS* = "CC00AA0A ".
Note: the quotes are not part of the message, but are used here to indicate that the space is part of the message.

There will be a runtime error (i.e. an error occurring during output creation), if the supplied message does not adhere to the format specified above.

Use the following options to configure the output Barcode settings:

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.

- **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.

- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the Barcode added to the page.
  - **Output once per sheet**: This option relates to Imposition printing (see Imposition), also known as N-Up printing. Select this box to have the Barcode printed once per sheet rather than once per document page.

    > **Note**
    >
    > If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

  - **Left**: Enter the distance between the left margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
  - **Bottom**: Enter the distance between the bottom margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.

- **Royal Mail 4-state Mailmark C Properties** group:
  - **Track height**: The height of the small vertical centre bars, in metric (mm/cm). Value must be between1.02 mm and 1.52 mm.
  - **Full bar height**: Height of the tallest vertical bars, in metric (mm/cm). Value must be between4.22 mm and 5.84 mm.

    The ascender and descender bar heights are automatically calculated based upon the *Full* and *Track* bar heights.
  - **Module width**: Thickness of each bar in the Barcode, in metric (mm/cm). Value must be between0.38 mm and 0.63 mm.
  - **Module spacing**: The spacing between each bar, in metric (mm/cm). Value must be between0.3 mm and 0.7 mm.
- **Process Tilde**:
  This selection is not applicable to this barcode type.

- **Text**: Enter the text used to generate the Barcode.
  - **Add** button: Click to display a list of variable data that could be used for generating the Barcode.
    This includes metadata fields which must previously have been added in the Metadata options, likely at the *Document Tags* level (see Metadata Options), as well as some information fields.
- **Condition**: Enter the condition which determines whether or not the Barcode will be added to the document at print time. For details on how to create a conditional, see "How to Set Conditionals" on page 772.
  - **Add** button: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

Royal Mail 4 State Mailmark L Settings

Royal Mail 4-State Mailmark L is used by the British Royal Mail postal service. The barcode contains an address, encoded in alpha-numeric format (0-9, A-Z).

The content (or message) of the barcode consists of the following 6 parts (or fields), which are detailed here:

Field	Length	Allowed Characters
Format	1	`[01234]`
Version ID	1	`[1234]`
Class	1	`[0123456789ABCDE]`
Supply Chain ID	6	`[0123456789]`
Item ID	8	`[0123456789]`
Destination Post Code plus DPS	9	`[0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ]` The value "XY11 " is used for this field to specify an *International Designation*.

As an example, the message `"11299999966448822CC00AA0A "` has *Format* = 1, *Version ID* = 1, *Class* = 2, *Supply Chain ID* = 999999, *Item ID* = 66448822 and *Destination Post Code plus DPS* = "CC00AA0A ".
Note: the quotes are not part of the message, but are used here to indicate that the space is part of the message.

There will be a runtime error (i.e. an error occurring during output creation), if the supplied message does not adhere to the format specified above.

Use the following options to configure the output Barcode settings:

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.
  - **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.
- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the Barcode added to the page.
  - **Output once per sheet**: This option relates to Imposition printing (see Imposition), also known as N-Up printing. Select this box to have the Barcode printed once per sheet rather than once per document page.

    > **Note**
    >
    > If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

  - **Left**: Enter the distance between the left margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
  - **Bottom**: Enter the distance between the bottom margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
- **Royal Mail 4-state Mailmark L Properties** group:
  - **Track height**: The height of the small vertical centre bars, in metric (mm/cm). Value must be between1.02 mm and 1.52 mm.

- **Full bar height**: Height of the tallest vertical bars, in metric (mm/cm). Value must be between4.22 mm and 5.84 mm.

  The ascender and descender bar heights are automatically calculated based upon the *Full* and *Track* bar heights.
- **Module width**: Thickness of each bar in the Barcode, in metric (mm/cm). Value must be between0.38 mm and 0.63 mm.
- **Module spacing**: The spacing between each bar, in metric (mm/cm). Value must be between0.3 mm and 0.7 mm.

- **Process Tilde**:
  This selection is not applicable to this barcode type.

- **Text**: Enter the text used to generate the Barcode.
  - **Add** button: Click to display a list of variable data that could be used for generating the Barcode.
    This includes metadata fields which must previously have been added in the Metadata options, likely at the *Document Tags* level (see Metadata Options), as well as some information fields.
- **Condition**: Enter the condition which determines whether or not the Barcode will be added to the document at print time. For details on how to create a conditional, see "How to Set Conditionals" on page 772.
  - **Add** button: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

UPC-A Settings

The Universal Product Code (UPC-A) Barcode is widely used for tracking trade items in stores and at the point-of-sale. It consists of 12 numerical digits which are uniquely assigned to each trade item.

Note

To create dynamic barcodes, Metadata options must first have been set (see "Meta Data options" on page 699).
Metadata fields are required to create the association between the dynamic data used in

the print run and the barcode.

Use the following options to configure the output Barcode settings:

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.
  - **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.
- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the Barcode added to the page.
  - **Output once per sheet**: This option relates to Imposition printing (see Imposition), also known as N-Up printing. Select this box to have the Barcode printed once per sheet rather than once per document page.

    Note

    If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

  - **Left**: Enter the distance between the left margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
  - **Bottom**: Enter the distance between the bottom margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.

UPC A Properties group:

- **Height**: Enter the Barcode height in either metric (cm/mm), inch (in) or point (pt) values.
- **Module Width**: Specifies the width of the narrow bars. Changing this value to a higher value will generally make the Barcode bigger.
    - **Checksum**: Use the drop-down to select how to deal with the Barcode checksum:
    - **Ignore**: Ignore checksum calculations.
    - **Auto**: Add a checksum character to the Barcode if the initial value does not validate. This is the default value.
    - **Check**: Verify the Barcode has a valid checksum.
    - **Add**: Calculate and add a checksum character to Barcode, regardless of current value.
    - **Print human readable text:** Check to add a textual version of the Barcode data.
    - **Placement**: Use the drop-down to select whether to place the human readable text above or below the Barcode.
    - **Font name**: Use the drop-down to select the font with which to display the human readable text.

      > **Note**
      >
      > Vertically-oriented fonts (fonts whose typeface name begin with the at (@) character) are not supported in Connect.

    - **Font size**: Enter a font size for the human readable text.
- **Process Tilde**:
  This selection is not applicable to this barcode type.
- **Text**: Enter the text used to generate the Barcode.
    - **Add** button: Click to display a list of variable data that could be used for generating the Barcode.
      This includes metadata fields which must previously have been added in the Metadata options, likely at the *Document Tags* level (see Metadata Options), as well as some information fields.
- **Condition**: Enter the condition which determines whether or not the Barcode will be added to the document at print time. For details on how to create a conditional, see "How to Set Conditionals" on page 772.

- **Add** button: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

UPC-E Settings

The Universal Product Code (UPC-E) Barcode is widely used for tracking trade items in stores and at the point-of-sale. It consists of 6 numerical digits which are uniquely assigned to each trade item.

Note

To create dynamic barcodes, Metadata options must first have been set (see "Meta Data options" on page 699).
Metadata fields are required to create the association between the dynamic data used in the print run and the barcode.

Use the following options to configure the output Barcode settings:

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.
  - **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.
- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the Barcode added to the page.
  - **Output once per sheet**: This option relates to Imposition printing (see Imposition), also known as N-Up printing. Select this box to have the Barcode printed once per sheet rather than once per document page.

> **Note**
>
> If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

- **Left**: Enter the distance between the left margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
- **Bottom**: Enter the distance between the bottom margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.

UPC A Properties group:

- **Height**: Enter the Barcode height in either metric (cm/mm), inch (in) or point (pt) values.
- **Module Width**: Specifies the width of the narrow bars. Changing this value to a higher value will generally make the Barcode bigger.
  - **Checksum**: Use the drop-down to select how to deal with the Barcode checksum:
  - **Ignore**: Ignore checksum calculations.
  - **Auto**: Add a checksum character to the Barcode if the initial value does not validate. This is the default value.
  - **Check**: Verify the Barcode has a valid checksum.
  - **Add**: Calculate and add a checksum character to Barcode, regardless of current value.
  - **Print human readable text:** Check to add a textual version of the Barcode data.
  - **Placement**: Use the drop-down to select whether to place the human readable text above or below the Barcode.
  - **Font name**: Use the drop-down to select the font with which to display the human readable text.

> **Note**
>
> Vertically-oriented fonts (fonts whose typeface name begin with the at (@)

character) are not supported in Connect.

- **Font size**: Enter a font size for the human readable text.

- **Process Tilde**:
  This selection is not applicable to this barcode type.

- **Text**: Enter the text used to generate the Barcode.
  - **Add** button: Click to display a list of variable data that could be used for generating the Barcode.
    This includes metadata fields which must previously have been added in the Metadata options, likely at the *Document Tags* level (see Metadata Options), as well as some information fields.
- **Condition**: Enter the condition which determines whether or not the Barcode will be added to the document at print time. For details on how to create a conditional, see "How to Set Conditionals" on page 772.
  - **Add** button: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

US Postal Service IMb Settings

US Postal Service Intelligent Mail Barcodes (IMb) are used by the United States postal service. They contain the address of the receiver, encoded in numeric format (0-9).

Note

To create dynamic barcodes, Metadata options must first have been set (see "Meta Data options" on page 699).
Metadata fields are required to create the association between the dynamic data used in the print run and the barcode.

Use the following options to configure the output Barcode settings:

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.
  - **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.
- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the Barcode added to the page.
  - **Output once per sheet**: This option relates to Imposition printing (see Imposition), also known as N-Up printing. Select this box to have the Barcode printed once per sheet rather than once per document page.

    > **Note**
    >
    > If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

  - **Left**: Enter the distance between the left margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
  - **Bottom**: Enter the distance between the bottom margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
- **USPS IMB Properties** group:
  - **Track height**: The height of the small vertical centre tracking bars, in inches (in). Value must be between 0.039 and 0.057 inches.
  - **Ascender height**: The height of the ascender bar.
    This also determines the height of the descender bar and full height bars, in inches (in). Value must be between 0.082 and 0.111 inches.
  - **Module width**: Thickness of each bar in the Barcode, in inches (in). Value must be between 0.015 and 0.025 inches.

- **Interchar gap size**: The gap between the bars, calculated as a ratio of the module width, where a value of 1.0 equates to a gap width the same as the module width. Value must be between 0.012 and 0.04 inches.
    - **Print human readable text:** Check to add a textual version of the Barcode data.
    - **Placement**: Use the drop-down to select whether to place the human readable text above or below the Barcode.
    - **Font name**: Use the drop-down to select the font with which to display the human readable text.

      Note

      Vertically-oriented fonts (fonts whose typeface name begin with the at (@) character) are not supported in Connect.

    - **Font size**: Enter a font size for the human readable text.

- **Process Tilde**:
  This selection is not applicable to this barcode type.

- **Text**: Enter the text used to generate the Barcode.
    - **Add** button: Click to display a list of variable data that could be used for generating the Barcode.
      This includes metadata fields which must previously have been added in the Metadata options, likely at the *Document Tags* level (see Metadata Options), as well as some information fields.
- **Condition**: Enter the condition which determines whether or not the Barcode will be added to the document at print time. For details on how to create a conditional, see "How to Set Conditionals" on page 772.
    - **Add** button: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

US Postal Service IMpb Settings

US Postal Service Intelligent Mail Package Barcodes (IMpb) are used by the United States postal service for parcel deliveries. They contain parcel destination information along with other

information about the delivery.

> **Note**
>
> To create dynamic barcodes, Metadata options must first have been set (see "Meta Data options" on page 699).
> Metadata fields are required to create the association between the dynamic data used in the print run and the barcode.

Use the following options to configure the output Barcode settings:

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.
  - **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.
- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the Barcode added to the page.
  - **Output once per sheet**: This option relates to Imposition printing (see Imposition), also known as N-Up printing. Select this box to have the Barcode printed once per sheet rather than once per document page.

    > **Note**
    >
    > If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

  - **Left**: Enter the distance between the left margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.

- **Bottom**: Enter the distance between the bottom margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.

- **USPS IMPB Properties** group:
  - **Barcode height**: The height of the bars in the barcode, in inches (in). Value must be between 0.075 and 1.0 inches.
  - **Module width**: The thickness of thin bars in the barcode, in inches (in). Thicker bars are a multiple of this value. Value must be between 0.013 and 0.021 inches.
  - **Text to barcode clearance**: The space between the barcode bars and the top and bottom text, in inches (in). Value must be between 0.125 and 0.25 inches.
  - **Text height**: The height of the text, in inches (in). Value must be between 0.09 and 0.125 inches.
  - **Line to text clearance**: The space between the guard lines and text, in inches (in). Value must be between 0.031 and 0.04 inches.
  - **Guard bar height**: The thickness of the guard lines, in inches (in). Value must be between 0.031 and 0.063 inches.

- **Process Tilde**:
  This selection is not applicable to this barcode type.

- **Text**: Enter the text used to generate the Barcode.
  - **Add** button: Click to display a list of variable data that could be used for generating the Barcode.
    This includes metadata fields which must previously have been added in the Metadata options, likely at the *Document Tags* level (see Metadata Options), as well as some information fields.
- **Condition**: Enter the condition which determines whether or not the Barcode will be added to the document at print time. For details on how to create a conditional, see "How to Set Conditionals" on page 772.
  - **Add** button: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

Additional OMR Mark Settings

The **Add OMR** dialog displays the properties of an OMR Mark that was added in the "Additional Content" on page 732 page.

These OMR marks differ from High Capacity Feeder (HCF) generated inserter marks. Those marks are specific to the inserter machine they were created for, whereas these additional OMR marks are completely independent and customizable. These custom OMR marks can be used to cater for inserter machines not currently supported by a HCF, or they can be used for any non-inserter related post-processing driven by OMR marks.

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.
  - **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.
- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the OMR Mark added to the page.
  - **Page Side**: Select whether the OMR Mark will print on the front or back of page.
  - **Output once per sheet**: Option relates to Imposition (also known as N-Up) printing. Select this box to have the OMR Mark printed once per sheet rather than once per document page.

    > **Note**
    >
    > If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

  - **Left**: Enter the distance between the left margin of the page and the OMR Mark, in either metric (cm/mm), inch (in), pixel (px) or point (pt) values.
  - **Bottom**: Enter the distance between the bottom margin of the page and the OMR Mark, in either metric (cm/mm), inch (in), pixel (px) or point (pt) values.
- **Options Tab**:

- **Collation Level**: Choices are:
  - **Document**: Treats each document as a group and the group and match marks will be set based upon the start and end of a document.
  - **Document Set**: Treats each document set as a group and the group and match marks will be set based upon the start and end of a document set.
- **Draw Hot Spots**: This adds a red rectangle around the location of each individual mark in the output, allowing easier checking of the OMR mark logic.
- **Line Options** group:
  - **Line Thickness**: Sets the thickness of each OMR mark line.
  - **Line Length**: Sets the lenght of each OMR mark line.
  - **Line Spacing**: Determines how the spacing between each OMR mark line will be set. The associated control beneath the combination box will be enabled, based upon this selection.
    - **Line Per Inch**: If **Line Spacing** is set to *Lines Per Inch* this option will be enabled. It defines how many lines will print per inch.
    - **Gap Distance**: If **Line Spacing** is set to *Gap Distance* this option will be enabled. It defines the size of the gap between lines; i.e. the distance from the bottom of one OMR mark line to the top of the next.
    - **Line Distance**: If **Line Spacing** is set to *Line Distance* this option will be enabled. It defines the distance from the top of one line to the top of the next.
- **Sequence Number Range** group: Allows selection of Start and Stop points for the wrapping page sequence number in a group.
  For example, a range of 2-10 would cause the sequence numbers to iterate as follows: 2, 3, 4, 5, 6, 7, 8, 9, 10, 2, 3, 4 ...

  > **Note**
  >
  > The Sequence Number iterates per page within a group and is used to identify missing pages in a group.

  - **Start**: The starting point for the range
  - **Stop**: The end point of the range

- **Start number**: The number to start from (from within the selected range).

- **Match Number Range** group: Allows selection of Start and Stop points for the wrapping match number for a group.
  For example, a range of 1-6, with a Start number of 2 would cause the matched numbers to be as follows: 2, 3, 4, 5, 6, 2, 3, 4, 5, 6, 2, ...

  > **Note**
  >
  > The Match Number iterates per group and is used to identify missing groups

  - **Start**: Start number
  - **Stop**: Stop number
- **Condition**: Enter the condition which determines whether or not the OMR Mark will be added to the document at print time.
  For details on how to create a conditional, see the "How to Set Conditionals" on page 772 page.

- **OMR Marks Tab**:
  - **#**: OMR Mark number (display only).
  - **Type**: Type of OMR Mark (display only).
  - **Value**: OMR Mark Value. These can be selected and altered for Sequence, Match and Parity marks, as described below.
  - **Add**: Add an OMR Mark entry to the table.
    Choices are between:
    - **On**: This represents a mark that is always printed.
    - **Off**: This represents a mark that is never printed; i.e. it pads the marks out with an empty position.
    - **Group Start**: This represents a mark that is printed on the first page of a group.
    - **Group End**: This represents a mark that is printed on the last page of a group.

> **Note**
>
> In a single page group both Group Start and End marks will print if defined since the page is both the start and end of the group.

- **Sequence**: This represents a mark that is printed when the specified bit is set in the sequence number of the page.
  For example, if the bit for the mark is set to 2 and the sequence number for the page is 5 then it will not print since the value 5 consists of the bits 1 and 4.
  Use the drop down box to select the entry.
- **Match**: This represents a mark that is printed when the specified bit is set in the match number of the group.
  For example, if the bit for the mark is set to 2 and the match number for the group is 3 then it will print since 3 consists of the bits 1 and 2.
  Use the drop down box to select the entry.

> **Note**
>
> The match number is the same for all pages in a group.

- **Parity**: This mark prints in order to maintain the parity of the number of lines printed on the page. If set to *Even* then it will print if the total count of the other printed marks in the printed is odd.
  For example, by printing the parity mark it will create an even number of marks on the page. And vice versa with *Odd* parity - the parity mark will print if the total number of other printed marks on the page is even in order to keep the overall count odd.
  Use the drop down box to select the entry.
- **Conditional**: Enter the condition which determines whether or not this OMR Mark will be added to the document at print time.
  For details on how to create a conditional, see the "How to Set Conditionals" on page 772 page

- ✖ **Delete**: Delete an entry from the table

- **Move up**: Move a entry up the table
- **Move down**: Move a entry down the table
- **Edit**: Edit a **Conditional** entry within the table.

> **Tip**
>
> You can also double click a **Conditional** entry within the table to edit it.

How to Set Conditionals

- **Condition**: Enter the condition which determines whether or not this element will be added to the document at print time.

  Use JavaScript conditional expressions to construct your conditional.
  You may use JavaScript logical operators (==, !=, <, >, &&, ||, etc), literal values ("text', 3.14, etc), mathematical operators (/, *, +, -, etc) and data fields.

  - **Add**: Click to add a field to the to the conditional expression.
    Select from a list of metadata fields (as added in Metadata Options), document information fields (see Variables Available in Output), or common expressions.

    For example, selecting the **Expressions > First Page** would add "page.nr == 1" to the conditional entry.

    Example: if (page.nr == 1) { ... }

Job Creation Presets Wizard

A Job Creation Preset is a settings file by which the Connect Server can filter, sort, and group print content items, add meta data and make finishing settings.

The **File>Print Presets>Job Creation Settings ...** dialog displays a list of available presets and a summary of their settings. This dialog can be used to create new Presets or to edit and update existing Presets. Presets, however, cannot be deleted or renamed from within this dialog. That must be done manually.

The Presets are all stored as individual files, using the Preset name and a "OL-jobpreset" file extension.

The Presets can be found in the following folder: `C:\Users\[UserName]\Connect\workspace\configurations\JobCreationConfig`
Where `[username]` is replaced by the appropriate Windows user name.

Actually, the path may not begin with 'C:\Users', as this is language-dependent. On a French system, for example, it would be 'C:\Utilisateurs'.

Type `%userprofile%` in a Windows File Explorer and press Enter to open the actual current user's home directory.

Once saved, Job Presets can be loaded in the **Print Wizard**, to have the Print Wizard use the selected Job values.

For more information about Job Creation Presets, see "Print Presets" on page 998.

> **Note**
>
> The **Job Creation** Wizard requires an active Template and associated data.

Dialog Interface

- **Data Mapping Configuration**: Use the drop-down to select which data mapping configuration this job creation preset will be based on. The data mapping configuration's model is used for field names in sorting, etc.
- **Configuration Name**: Use the drop-down to select the presets saved in the default location Click the Gear icon for more options:
  - Click the **Reload** option to look for new presets.
  - Click the **Import Configuration...** option to import one or more Job Presets using a Browse dialog.
- **Properties**: Displays a summary of the settings for this Job Creation Preset.
  - **Has Custom Job Creation Options**: Indicates if any job creation settings have been added. Becomes Yes if any setting in any of the below windows have been

added:

- **Has Data Selection Filter**: Becomes Yes if the "Data filtering options" on page 689 are set.
- **Has Sorting**: Becomes Yes if any "Sorting options" on page 693 are set.
- **Has Grouping**: Becomes Yes if grouping options are set in the "Grouping options" on page 696.
- **Page Count Splitting**: Becomes Yes if page count splitting is used in the "Grouping options" on page 696.
- **Slip Sheets**: Becomes Yes if a slip sheet is set in the "Grouping options" on page 696.

- **Options**: These options are checked, or not, depending on the selected preset chosen in the Configuration name.
  - **Runtime Parameters**: select "Runtime Parameter Options" below to store information which can be used at runtime for comparisons against conditions within Job Creation, or for use with external sorting programs.
  - **Use Grouping**: Check to configure grouping of output into jobs, job segments or document sets. See "Grouping options" on page 696.
  - **Apply filtering and sorting to record selection**: Check to activate the "Data filtering options" on page 689 (to filter out certain records) and "Sorting options" on page 693 (to sort the remaining records) pages of the wizard.
  - **Include Metadata** (only available for *PDF* and *AFP output*): Check to add meta data to the output. This can be done at Job, Job Segment, Document, Document Set and Page level. See "Meta Data options" on page 699.
  - **Override Finishing Options**: Check to configure custom "Finishing options" on page 691, such as binding.
- **Next**: Click to go to the next page of the Job Creation Wizard, "Data filtering options" on page 689.
- **Finish**: At any point during the wizard, click to save the current configurations, whatever page you are on.
- **Cancel**: At any point during the wizard, click to exit the wizard without saving changes.

Runtime Parameter Options

This page appears as part of the **Advanced Print Wizard** and the "Job Creation Presets Wizard" on page 684.

The **Runtime Parameter Options** page allows you to add runtime parameters to the Job.

These runtime parameters could be used in a number of ways. Such as in the following scenarios:

Filtering rules scenarios

A filter condition could be used to compare a date from the record against a date passed in as a runtime parameter.

For example, a list of document types for a job. The document could have a property with its document type (e.g., reminder, collection letter, etc.), and the runtime parameter could be a comma separated list of document types for the job. So a rule would then compare if document type occurs in the list of document types.

Finishing rules scenarios

Similar to filtering, a finishing condition could use runtime parameters to compare against.

For example, consider if a copy of a print job has to be produced, in which the copies have to be stapled, while the original print run are without staples. A runtime parameter could be used during job creation to indicate if the Job Creation is working on the original or the copy.

Without runtime parameters, the same functionality would require two job presets. So this use case becomes important for relatively complex presets where duplicating the preset creates a maintenance issue.

External sorting scenarios

External sorting benefits considerably from runtime parameters.

Especially in case of postal sorting, there can be a need for parameters that vary per job, such as intended delivery date, tracking id, and more.

Metadata properties

The value for a metadata property could be supplied through a runtime parameter. These could be, for example, the production date, or the name of the current operator.

Page breakdown

Runtime Parameters can be added, edited, removed, duplicated or moved within the Runtime Parameter table via the buttons to the right of the table.

The **Runtime Parameter options** are:

- **Add**: Click to open a new Runtime Parameter.
  This launches the Add/Edit Runtime Parameter dialog.
- **Edit**: Click to edit the currently selected entry. This launches the Add/Edit Runtime Parameter dialog.
  Double clicking on an entry in the table has the same effect as this button.
- **Delete**: Click to delete the currently selected entry or entries.
- **Duplicate**: Click to create a duplicated copy of the entry. This creates a new entry that is exactly the same as the original, with a numeric increment added to the end of the name.
- / **Move Up / Move Down**: Click to move the selected entry (or entries) up or down within the table.

Runtime Parameter Table: The table lists all the currently available runtime parameters.
It is split up into the following columns

- **Parameter**: The name of the parameter.
- **Type**: The parameter type.
- **Default Value**: Displays the default parameter value.

Advanced Print Wizard and Output Creation Preset Wizard navigation options

- **Preview** button (active in ***Advanced Print*** Wizard only): Click to launch a Proof Preview window, which displays how the printed output would look based upon the currently chosen selections.
- **Back** and **Next** buttons: Used to navigate back and forth through all the selected options within the Wizard. Up until the Print button is pressed, one can reverse all the way through the wizard to return to the main selection page (the "Print options" on page 704 page) and add or remove printing options from the print run.
- **Print** button (***Advanced Print*** Wizard only) or **Finish** button (***Output Creation Preset*** Wizard only):
  Click to produce print output/finalize the Preset according to the current settings.
  This can be done at any point within the Wizard, whether or not *all* the options selected in the "Print options" on page 704 page have been completed or not.
- **Cancel** button: Cancels the Print Wizard, without creating any printout.

Data filtering options

This page appears as part of the **Advanced Print Wizard** and the "Job Creation Presets Wizard" on page 684.

The **Data Filtering Options** page is used to filter records to prevent them from being printed. Conditions are evaluated on each record, and can be based upon such diverse data as the data fields used in records or the physical printing options for a record.

The rule(s) apply at a Group level. There may be one rule or many rules at the same level, and there may be groups within groups, providing the ability to create quite complex nested logical structures.

Group

A group consists of one or more rules with a logic operator. Four logic choices are available at the Group level. These are:

- **All of the following**.
  This equates to the logical operator (... AND ...).
  If *all* of the associated criteria are met, then this group resolves to TRUE.
- **Any of the following**.
  This equates to the logical operator (... OR ...).
  If *any* of the associated criteria are met, then this group resolves to TRUE.
- **Not all of the following**.
  This equates to the logical operator (NOT(... AND ...)).
  If *any (but not all)* of the associated criteria are met, then this group resolves to TRUE.
- **Not any of the following**.
  This equates to the logical operator (NOT (... OR ...)).
  If *none* of the associated criteria are met, then this group resolves to TRUE.

The top level is always a group. A group can contain one or more Rules and/or Groups.

Rule

A Rule is a logic expression using a single operator or function, and their associated operand (s). The number and type of operands is dependent upon the operator or function chosen.

Rules can also have different types. The available rule types depend upon the context where the Rule Editor is used. For instance, within Job Creation, there can be data rules that are

based on record data belonging to a content item, but there will also be "location rules" that deal with the position of a document within the job.

The Rule types are as follows:

- **Data Rule**: This rule operates upon Data entries. Select the Data Field, then select the logical test and the associated test value(s). The options available will depend upon the Data Field type (they are "type aware"), but all have at least "*is equal to*" and "*is not equal to*" to another entry as an option. The "*is set*" and "*is not set*" selections allow to determine whether the data field has a value or not (i.e. do a null comparison).
  The data type specific options are as follows:
  - **Alphanumeric** data field specific selections: Whether the alphanumeric string "*contains*", "*does not contain*", "*starts with*" or "*ends with*" another string value.
    The "*is like*" and the "*is not like*" selections allow the use of "*" wildcards.
    The comparison string could be a manually entered string, or it could be another textual data field. The data field could be from the data model, or it could be a runtime parameter entered as part of Job Creation. Runtime parameters are expressed as "params.xxxx", where the xxxx is the name of the runtime parameter.

    In the Data Filtering Options rules editor the comparisons are always case sensitive. In other rule editors there is an option to turn off case sensitivity.
  - **Date** data field specific selections: Whether the date field is "*before*", "*after*", "*not before*" or "*not after*" another date entry, or "*is between*" two other date entries. The comparison dates can be a manually entered dates, or date data fields (either from the data model or runtime parameters).
    The date value should be in ISO 8601 format. When entering dates manually it is best to select the date using the date selection option.
  - **Boolean** data field specific selections. Check whether the selected Boolean data field "*is true*" or "*is false*", or whether it "*is equal to*", or "*is not equal to*" a Boolean runtime parameter.
  - **Numeric** data field specific selections: Whether the numeric fields is "*less than*", "*less than or equal to*", "*greater than*" or "*greater than or equal to*" another number entry. The comparison number could be a specified number, or it could be another numeric data field (either from the data model or runtime parameters).
- **Media Rule**: This rule operates on either the Media *Name* or coating (either *Front Coating* or *Back Coating*) .
  For the *Name* value, the choice is a straight binary option based upon the selected Media Name.

For the *Coating* values, the choice is about what type of coating applies (with an "Unspecified" option as a fall back).

- **Binding Rule**: This rule operates upon the binding that applies to the record. The options are based on the *Style*, *Side*, *Location* or *Angle* of the binding. The logical operators that apply are dependent upon the type of Binding the test is being applied to.
- **Duplex Rule**: This rule operates upon the record Duplex settings. This is a binary choice between Duplex or non-Duplex.
- **Size Rule**: This rule operates upon the size of the selected group. This could be the amount of sub-groups contained within, or the amount of physical *Sheets*, *Pages* or *Sections*. The logical operators that apply are dependent upon the group that the size test is being applied to.
- **Property Rule**: This rule compares a job *Property* value (such as "Control Scripts" on page 429 set within Designer) against a selected string. The comparison operators that are available for this rule are the same as those for Alphanumeric **Data Rules**. The only difference is that the "*is set*" and "*is not set*" selections not only check for null property values but also for properties that don't exist at all.

Preview

This box displays a textual representation of the logical conditions set within the data filtering table.

Finishing options

This page appears as part of the **Advanced Print Wizard** and the "Job Creation Presets Wizard" on page 684.

Use the **Finishing Options** page to force the use of specific printer finishing options, rather than using any finishing options that might have been set in the template's Print Context and Print Section options.

These settings are only applied when producing print output. They do not modify the original finishing options in either the Section or the Template.

Finishing settings can be set on various levels of Job Creation, such as Document Sets and Job Segments. This allows you, for example, to staple Document Sets whilst punching holes in Job Segments.
You can also have multiple finishing settings at the same level.

- **Finishing Options Table**: This table, on the left of the Wizard page, allows you to add or remove sections to apply the Finishing options to.
  It contains the **Section** for which the settings are to be applied (*Section*, *Document*, *Document Set* or *Job Segment*), plus the **Finishing** options that are to be applied, and any **Rule** determining the conditions under which the Finishing settings are to be applied.
  To set the details, use the following options:
  - **Import Finishing:** Imports the existing settings from a Connect Template. The Template could be the current template (***Import from current template***), or some other Template (***Import from template ...***).
  - **Add:** Adds a new section to apply Finishing options to. The default is to add a new *Section*, but these can be changed to *Document*, *Document Set* or *Job Segment* types, thereafter.
    You can have multiple selections of each type, all with different criteria determining when the Finishing options are to be applied.
  - **Delete:** Removes the selected Section(s) from the table.
  - **Move Up / Move Down:** Moves the selected Section(s) up or down within the table.
  - **Edit ... :** Brings up the Rule Editor Dialog, which can be used to construct the rule (s) that determines whether this conditional Finishing option is to be applied or not.
- **Binding** group:
  - **Style**: What type of Binding to request on the printer. This includes *Stapled, Glued, Stitched, Ring, Comb, Coil*, amongst other options..
  - **Side**: Sets the side of the paper that the Binding is to occur. This includes both the top and the bottom of the paper,
  - **Location**: Sets where the binding is to occur, if applicable.
    The selections available here will be dependent upon the selection made in the Binding **Style**. Only Stapled and Stitched bindings have a **Location** option available to them.
  - **Angle**: Set Stapling or Stitching binding either horizontally, vertically, or at an angle (as supported by printer).
  - **Item count**: Select the amount of Staples or Stitches to use. The choice is between the default amount or selecting a specific number using the Count option.

> **Tip**
>
> The options available to you in reality at print time will be printer dependent, so you will need to know the capabilities of your printer, or leave the value set to Default.

  - **Area**: The area where the binding can be applied.
- **Hole making** group:
  Hole making options are available only to *Ring, Comb* (wire and plastic) and *Coil* Binding **Styles**. The selections will need to be made at run-time based upon the types of binding options available that the printer supports.
  - **Number of holes**: The number of holes to punch for the selected Binding option.
  - **Pattern Catalog ID**: The Catalog ID of the selected Binding option.

Advanced Print Wizard navigation options

- **Preview** button: Click to launch a Proof Preview window, which displays how the printed output would look based upon the currently chosen selections.
- **Back** and **Next** buttons: Used to navigate back and forth through all the selected options within the Wizard. Up until the Print button is pressed, one can reverse all the way through the wizard to return to the main selection page (the "Print options" on page 704 page) and add or remove printing options from the print run.
- **Print** button: Click to produce print output according to the current settings. This can be done at any point within the Wizard, whether or not *all* the options selected in the "Print options" on page 704 page have been completed or not.
- **Cancel** button: Cancels the Print Wizard, without creating any printout.

Sorting options

This page appears as part of the **Advanced Print Wizard** and the "Job Creation Presets Wizard" on page 684.

The **Sorting Options** page is used to sort the records in the output. Sorting is done from the top to the bottom, one after the other.

Sorting Settings

- **Use standard sort**: Sort using the fields below:
  - **Field Name**: Use the drop-down to select which field to sort on.
  - **Order**: Use the drop-down to choose Ascending or Descending.
  - **Add**: Click to add a new row to the sort list. The list that appears contains all the fields in the Data Model, as well as a special <Document Length> option which is used to sort by the number of pages in each document.
  - **Delete**: Click to delete the currently selected row in the list.
  - / **Move up / down**: Click to move the currently selected row up or down within the list.
- **Use external sort**: Sort the records using some external sorting software. A CSV file is exported, then sorted by the external application and the resultant sorted CSV file is returned and integrated, with all the records now being sorted according to the new order within the CSV file.

> **Warning**
>
> ***External Sort commands must return a non-zero error code if an error occurs.***
>
> An external sort command could easily fail part way through processing and generate only a partial output file. Without receiving a return code from the external sort process, PrintShop Mail Connect cannot know if the sort has successfully completed or not. Thus the sort program must generate a return code, with a code of zero ('0') indicating success, and all non-zero results indicating failure.

- **External Sort Settings** group:
  This section of the dialog is only activated if the **Use external sort** option has been selected.
  - **Command**: Enter either the full path of the executable file that will sort the CSV file, or a valid Windows command line instruction to sort the records.

    Placeholders can be used within this command, and are available via the **Insert input/output file placeholder** options (expressed as either `${input}` or `${output}`) or via Job Creation runtime **Parameters** (expressed as `${params.xxxx}`, where the xxxx is the name of the runtime parameter).

    The Windows command line instruction should do something like the following:

1. Do some processing of the input CSV file which PrintShop Mail Connect will pass through in the position of the `${input}` placeholder.
2. Generate an output file that contains the sorted data and must be named according the file name PrintShop Mail Connect will pass through in the position of the `${output}` placeholder

For example: `cmd /C sort /R ${input} ${output}`
This would reverse the order of the `${input}` file, and sent the output to the `${output}` file.

- **Timeout**: Enter the number of seconds to wait for an external sort command to complete before abandoning the external sort.
- **Separator**: Enter the field separator used in the CSV file, such as a comma (,), pipe (|), semicolon (;), etc.
- **Quote Character**: Enter the quoting character that wraps around any field that contains the separator.
- **Escape Character**: Enter the character use to escape the Quote character if it appears in the field value.
- **Line Ending**: Use the drop-down to select which line ending to use. The selections are: Windows *Carriage Return/Line Feed* combination (CRLF), Linux *Line Feed* (LF) or Apple Macintosh *Carriage Return* (CR).
- **Character Set**: Use the drop-down to select which character set to use when encoding the CSV file. This always defaults to UTF-8, as this caters for all possible characters, is relatively compact (in terms of Unicode character sets) and is compatible with standard ASCII.
- **Exported sort data** group:
  - **First row of sort data has field names** checkbox: select to have field names placed on the first line of the exported CSV file.
  - **Fields to export**: Lists the fields to export in the CSV file. The buttons to the right of the table provide the following functionality:
    - Click to select from available datatfields. The Field Selection dialog will appear, which allows selection of one or several fields from those available.
    - Click to remove a field from the list.
    - / Click to move fields up or down in the order of output.

- **Record ID Field**: The Record ID field is a database *Primary Key* field, which is automatically added to the exported data file. The Record ID field name defaults to *RecID*, but can be changed here as desired.

- **Processing returns data** group: This allows the external sort application to introduce new data for each record. This data can be embedded in the metadata and used as the source for additional content within PrintShop Mail Connect. An example usage would be generating a postcode or postal barcode data from address details, making it available for use in PrintShop Mail Connect.
  - **Processing returns data** checkbox: Select this if the sort processing will be returning data. This activates the whole optional **Processing returns data** subsection.
  - **First row of return data has field names** checkbox: select to have field names placed on the first line of the returning datafile.
  - **Return Fields**: Lists the fields available in the selected data mapping configuration that can be used to sort the records. Fields can be added or removed by use of the add datafield ( ) and remove datafield ( ) buttons, or re-arranged with the arrow buttons (/).
    Field names can be altered by selecting the field in the table, and editing the name.
    Fields can be made available to PrintShop Mail Connect via the "Include in meta data" checkbox. Click the checkbox beside the field name to make that datafield available as meta data.
  - **Record ID Field** selection box: Select which return field is to be the Record ID field.
  - **Sorting by** selection box: Select whether the sorting will be by the returned sort order or whether it is to be sorted on a selected datafield.
    - **Sequence Field** selection box: Select the datafield to be sorted on, if such was chosen in the **Sorting by** entry.

Grouping options

This page appears as part of the **Advanced Print Wizard** and the "Job Creation Presets Wizard" on page 684.

The **Grouping options** page separates the job output into multiple blocks that can then be physically separated using split sheets in the printer.

A typical usage of this feature is to create groups of mail pieces by size, so they can easily be

inserted (into envelopes). For instance, one might want to group together all the "single page" mail pieces that fit within a C5 envelope and then put all larger mail pieces into an "oversized" category, that will go into larger envelopes.

- **Grouping Tabs**: Jobs can be grouped at three different levels. The three groups each have their own tab, and are as follows:
  - **Job Grouping Fields**
  - **Job Segment Grouping Fields**
  - **Document Set Grouping Fields**

  The Fields available to be used for any Grouping are contained within the **Available Fields** box.

  Any Fields that you want to use for Grouping need to be added to the **Selected Fields** box via the arrows found between the two boxes.
  Simply select the Field(s) you want to move and then click the appropriate arrow.

  Any fields that you decide don't need to be used in Grouping can be returned to the Available Fields box in the same fashion.

  Once a field is added to the **Available Fields** box, its *Sorting Option* can be selected by clicking in the "**Sorting Option**" column, and selecting the appropriate option. The options for sorting are either Ascending or Descending order.

- **Size Grouping** section: Check the checkbox to enable *size grouping*, which separates Documents into different groups, based upon their relative sizes.

  For example, selecting *Document Set Grouping* for the **Grouping Level**, *Documents* for the **Item to group**, **Group** by *Page Count* and then creating a page range from 1-5 and another page range 6 to Largest, will create two Document Set groups. The first will contain all Documents of 1 to 5 pages in length, and the second will contain any document of 6 or more pages.

  The options within Size Grouping are:

  - **Grouping Level**: Use the drop-down to select which grouping level to use, between **Job**, **Job Segment** or **Document Set**. Only one grouping level can be selected.
  - **Item to Group**: Use the drop-down to select which item to group. The item choices are **Job Segments**, **Document Sets** or **Documents**.

> **Note**
>
> A selection made here can over-ride a previously selected **Grouping Level** option.
>
> For example, if the **Grouping Level** was previously set to *Document Set*, then selecting *Job Segments* as the **Item to Group** will not make sense for that Grouping Level. Thus the **Grouping Level** would automatically be changed to the more appropriate *Job* selection.

- **Group by**:Use the drop-down to select what criteria the Items will be grouped by. The choices are **Page Count** or **Sheet Count**.
- **Size groups in reverse order** checkbox: Reverses the order of the groups created. By default, grouping is done from smallest to largest. Checking this option instead creates groups from largest to smallest.
- **Size grouping after normal grouping** checkbox: Check this option to firstly group using the selections made above and then secondly group upon page break grouping. This creates two different levels of grouping, applied in order.
- **Meta data property** edit box: Select a name for the *meta data* field that is created. This meta data field can then be used in other Print Wizard/Preset pages, such as in the Job Output Mask Dialog.
- **Size Ranges**: Add (or remove) entries to this list to create new groups based upon the number of pages in the level selected above. All groups must be contiguous from 1-to-Largest and they must not contain any gaps.
  - **Range Name**: Enter a name identifying the range. It must be unique, but otherwise bears no impact on the range feature.
  - **From**: Enter the starting page number of the range. The first range must start with 1, all other ranges must be contiguous (the "From" range must be one higher than the previous "To" value).
  - **To**: Enter the last page number for the range. The last range must end with a selection of "Largest".

Advanced Print Wizard navigation options

- **Preview** button: Click to launch a Proof Preview window, which displays how the printed output would look based upon the currently chosen selections.

- **Back** and **Next** buttons: Used to navigate back and forth through all the selected options within the Wizard. Up until the Print button is pressed, one can reverse all the way through the wizard to return to the main selection page (the "Print options" on page 704 page) and add or remove printing options from the print run.
- **Print** button: Click to produce print output according to the current settings. This can be done at any point within the Wizard, whether or not *all* the options selected in the "Print options" on page 704 page have been completed or not.
- **Cancel** button: Cancels the Print Wizard, without creating any printout.

Meta Data options

This page appears as part of the **Advanced Print Wizard** and the "Job Creation Presets Wizard" on page 684.

The **Meta Data Options** page defines meta data entries that will be added to PDF output files and optionally added to AFP output (where they are added as TLEs by default, but can be removed in the "AFP Options" on page 709 page).

Meta data entries are ignored in all other output types, except when they are associated with "Additional Content" on page 732 or used as conditionals or in output file names, in the Advanced Print Wizard or in "Output Creation Presets" on page 1001.

> **Note**
>
> Meta data properties have a length limitation of 255 characters.

The meta data entries can be added to any of these levels, as indicated by the tabs on top: **Job**, **Job Segment**, **Document**, **Document Set**, and **Page** Tags.

> **Note**
>
> It is important to note that if Impositioning is used then the only meta data that will be available to the job thereafter will be the meta data at Job Segment level.
>
> When using Impositioning everything within the Segment is Impositioned, so all meta data below the Job Segment level is lost.

In each of these levels, a list of entries is available in a table. The Table options are as follows:

- **Always create meta data for this level even when fields are selected**: Select this check box to create a blank meta data entry if no fields are selected. Done to ensure that a meta data store is always available, if required.
- **Tag Name** column: Name of the meta data tag added to this level. Once a tag has been added, its name can be edited by double-clicking on the Tag Name.
- **Source Type** column: Displays the type of field being used - either Text or Data Field.
- **Source** column: For Data Fields and runtime parameters only. Either the runtime parameter name or the Field name from the data mapping configuration whose value will be used for this tag.
- **Add meta data**: Click to add a new tag to the current level. Select from one of **Add field meta data**, **Add runtime parameter meta data** or **Add text meta data** from the Field Selection context menu. Those options do the following:
  - The **Add field meta data** option launches the Field Selection dialog.
  - The **Add runtime parameter meta data** option launches the Runtime Parameter Selection dialog.
  - The **Add text meta data** option inserts a new text meta data entry straight into the current meta data table.

  Once loaded in the table, the meta data fields *Tag Names* can be modified as desired. As can the source they point to, in the case of runtime parameters and data fields.
- **Delete**: Click to delete the currently selected meta data entry.
- / **Move Up / Move Down**: Click to move the currently selected tag(s) one position up or down.

Output Creation Presets Wizard

The **File > Print Presets > Output Creation Settings...** dialog displays a list of available presets and a summary of their settings. This dialog can be used to create new Presets or to edit and update existing Presets. Presets, however, cannot be deleted or renamed from within this dialog. That must be done manually.

The Presets are all stored as individual files, using the Preset name and a "OL-outputpreset" file extension.

The Presets can be found in the following folder: `C:\Users\[UserName]\Connect\workspace\configurations\OutputCreationConfig`
Where `[UserName]` needs to be replaced with your own Windows User Name.

Once saved, Output Presets can be loaded in the **Print Wizard**, to have the Print Wizard use the selected Output values.

Dialog Interface

Configuration Selection section:

- **Configuration Name**: Use the drop-down to select the presets saved in the default location.
  Click the Settings button for more options:
  - Click the **Reload** option to look for new presets.
  - Click the **Import Configuration...** option to import one or more Output Presets using a Browse dialog.
    If Output Presets of the same name already exist in the Output Presets list, then a choice as to whether to overwrite the existing Preset(s) or not will be provided.
- **Properties**: Displays a summary of the settings for this Output Creation Preset.
  - Output Type: Displays the print technology used, as defined in the "Print options" on page 704.
  - Inserter: Indicates whether Inserter Marks have been added in the "Inserter options" on page 783 dialog. Expand to see which High Capacity Feeder (HCF) model is loaded.
  - Imposition: Indicates if Imposition has been set in the "Imposition options" on page 774 dialog. Expand to see the specific imposition settings.
  - Has custom printer settings: Indicates if custom printer settings have been set in the "Printer settings" on page 786 dialog. Expand to see the list of settings.
  - Output to: Indicates where the output will be done, either to a file or a printer.
  - Has Custom Finishing: Indicates that the output creation settings contain custom finishing overrides. (See "Job Creation Presets Wizard" on page 684.)

Optional Job Creation Configuration section:

- **Configuration Name**: Use the drop-down to select a preexisting Job Preset that can then be used as the source of Metadata and Grouping information in the output file mask dialog.
  Click the Reload button to refresh the list of available configurations.

Click Next in this dialog to see the "Print options" on page 704 window where output creation settings are selected.

Printer settings

This page appears as part of the **Advanced Print Wizard** and the Output Creation Preset Wizard.

The **Printer Settings** page provides options for cut-sheet printers. It maps media types to printer trays.
It is available for AFP, IPDS, PCL and PostScript printers that are configured for cut-sheet printing.

> **Note**
>
> Not all the settings will be available for all printers. Only those printers whose Printer Definition supports the extra information will have the **Position**, **Weight**, **Type** or **Color** options available.

- **Map media by** options: Select from following choices:
  This entry will only be available for those printers that support the extra information.
  - **Media Attribute** displays all Media details, except the Tray selection.
  - **Tray** displays just the Media name and Tray selections.
  - **Both** displays all Media details.
- **Media/Tray Table** columns:
  - **Media**: Lists the Media name, as defined in the template.
  - **Tray**: Use the drop-down to select in which Tray to send any page using the media.
  - **Position**: Enter a MediaPosition option on the printer to define the media to use.
    This entry will only be available for those printers that support the extra information.
  - **Weight**: Enter a weight for the paper.
    This entry will only be available for those printers that support the extra information.
  - **Type**: Use the drop-down to select which type of stock to use on the printer.
    This entry will only be available for those printers that support the extra information.
  - **Color**: Use the drop-down to select which color the paper should be on the printer.
    This entry will only be available for those printers that support the extra information.
- **Media/Tray Table** buttons:

- **Add**: Adds a new Media/Tray entry in the table.
- **Delete**: Deletes the current Media/Tray selections from the table.
- / **Move Up / Move Down**: Move the selected Media/Tray entries up or down within the table.
- **Import Tray Settings**: Import the Media/Tray settings from a Connect Template.

  This entry will only be available for those printers that support the extra information. The options are to:

  - **Import from current template**, which will import any Media/Tray settings from the current Connect Template into the table.
  - **Import from template file**, which allows you to browse for a Connect Template to import the Media/Trays settings from.

  There is no restriction on how many Templates you may import settings from.

- **Output Destination** group: These settings allow for the selection of output bins. It is available for AFP, IPDS and PCL printers whose Printer Definition caters for multiple output bins. If the selected Printer Definition does not support bins, this group will not be visible on the page.

  The table has the following choices:

  - **Trigger**: Select what the trigger will be for changing bins. The options are *>Page*, *>Sheet*, *>Document*, *>Document Set* or *>Job Segment*.
  - **Destination**: Select which bin should be used when this trigger is met. The options will be the bins available to the selected Printer Definition.
  - **Rules**: Defaults to true, but can be altered at will. Double clicking in the field brings up the Rule Editor, which can be used to construct the rule(s) that determines whether this bin is to be used or not.

  The options (both button and right mouse click context menu) available to the table are:

  - **Add**: Adds a new bin selection choice to the table.
  - **Delete**: Deletes the current selection(s) from the table.
  - / **Move Up / Move Down**: Move the selected entries up or down within the table.
  - **Edit ...**: Edit the current selection.

  Default output destination selection: Select the default output bin from those bins available to this Printer Definition.

Advanced Print Wizard and Output Creation Preset Wizard navigation options

- **Preview** button (active in ***Advanced Print*** Wizard only): Click to launch a Proof Preview window, which displays how the printed output would look based upon the currently chosen selections.
- **Back** and **Next** buttons: Used to navigate back and forth through all the selected options within the Wizard. Up until the Print button is pressed, one can reverse all the way through the wizard to return to the main selection page (the "Print options" below page) and add or remove printing options from the print run.
- **Print** button (***Advanced Print*** Wizard only) or **Finish** button (***Output Creation Preset*** Wizard only):
  Click to produce print output/finalize the Preset according to the current settings.
  This can be done at any point within the Wizard, whether or not *all* the options selected in the "Print options" below page have been completed or not.
- **Cancel** button: Cancels the Print Wizard, without creating any printout.

Print options

The **Print Options** page is the first page of the **Advanced Print Wizard** (used for both Production and Proof printing) and the second page of the "Output Creation Presets Wizard" on page 700.

This page is the most important of both the Advanced Print Wizard and the ***Output Creation Preset*** Wizard. All other pages that appear in the Wizard are determined by the selections made on this page.

The options that don't appear in the ***Output Creation Preset*** Wizard (as noted below), do appear in the ***Job Creation Preset*** Wizard. Only the Advanced Print Wizard contains all the options, but it cannot save your settings to file like the preset wizards allow.
For more information about print preset files, see "Print Presets" on page 998.

The choices can be broken down as follows.

Printer

- **Model**: Use the drop-down to select the printer language / output type that will be generated.
  Connect comes with several bundled print output Models which cover a large range of industry standard print output types.
  These include AFP, IPDS, PCL, PDF and PostScript (including the optimized PPML,

VIPP and VPS variants), with a range of quality settings available for each.

The bundled print output Models do not cater for *all* types of printers, however. There are simply too many different printers with their own individual (often quirky) settings and capabilities to cater for all of them with just a few standardized print Models. To use Connect with a particular printer Model you must obtain a tailored Connect Printer Definition that defines that printer and its capabilities for Connect. For example, to cater for IPDS or AFP printers that support Encapsulated Postscript resources included through an Include Object (IOB) structured field. Such a print output Model would need to be customized for you. These customized Printer Definitions will be created by Upland OL Support, in conjunction with information you provide about both your printer and your specific needs. Support will then provide you with a customized Printer Definition that supports the desired functionality with your specific printer.

By default, Connect initially displays just a single generic *PDF output* option, to keep the interface clean. But other print output types can be added to the Printer Model drop-down list at any time via the **Import Definitions** button

The **Import Definitions** options are:

- **Import Definition**: Used to add any customized or tailored Printer Definition that Upland Objectif Lune has prepared for you, specific to your printer.
- **Create Definition from PPD**: Used to create your own customized PostScript Printer Model through the printer manufacturer provided PostScript Printer Definition (PPD) file. This option launches the "Dynamic PPD Options" on page 788 page later in the Printer Wizard, which allows you to setup the printer.
- **Edit available printers**: Add extra Printer Definitions from the selection that come installed with Connect.
- **Export printer definition from output preset**: Use to extract a Printer Definition from a new or modified Printer Model. This option is only available when the Printer Model currently loaded differs from those previously saved or installed.

For more information on how to add Printer Definitions to Connect, see "Adding print output Models to the Print Wizard" on page 1005.

Output options

- **Output Local**: Select to have the output created using the local Print Server.
- **Output Type** choices:
  - **Prompt for file name**: Select to output to a local file on the hard drive. When this option is selected, no other configuration is necessary. A Save As dialog will appear to allow selection of the folder and file name.
  - **Directory**: Select to output to a local folder on the machine.
    Selecting this will open the **Directory Options** sub-group, which has these options:
    - **Job Output Mask**: The name of the file that will output.
      You could write the Job Output Mask directly into this edit box (for a list of available variables see "Print output variables" on page 1008), or you could create a Mask via the Options button. This opens the custom dialog: Job Output Mask Dialog.
      The Job Output Mask may contain (dynamic) folder names, for example:
      `${document.metadata['Country']}\${template}.`
      The evaluated value of the Job Output Mask is taken as a path **relative** to the folder specified by the Job Output Folder (the next option in this dialog). The Job Output Folder must exist, but folders specified in the Job Output Mask will get created if they don't exist.
    - **Job Output Folder**: The path on the disk where the file is produced. Please note that the folder must exist, or output will fail when produced through the server.
  - **LPR Queue**: Select to send the print job to an LPR queue. It is assumed that the print technology is supported by the system receiving the LPR job.
    - **Local Printer**: The IP or host name of the printer or machine where the LPD is installed and will receive
    - **Queue Name**: The queue name that will accept the job on the LPD. Default is generally "auto".
    - **Job Owner Name**: Optional entry for adding the name of the job owner.
    - **Job Name**: The name of the output file. You can use ${template} as a variable for the name of the Designer Template used to generate the output. (See also: "Print output variables" on page 1008.)
  - **Windows Printer**: Select to send the Print Job to a printer queue (note that this is an actual printer queue, not a Workflow Printer Queue).
    For non-PDF output, the job will be first rendered as a PDF before being printed through the Windows driver.

For PDF output the Adobe Library is used.

- **Windows Printer**: Use the drop-down to select the Windows printer queue where the job will be sent.
- **Job Owner Name**: Optional entry for adding the name of the job owner.
- **Job Name**: The name of the output file. You can use ${template} as a variable for the name of the Designer Template used to generate the output.
- **PDF Rendering Options** sub-group (only available for *PDF output*):
  - **Auto-rotate and center**: Check to automatically select the page orientation that best matches the content and paper.
  - **Choose paper source by page size**: Check to use the PDF page size to determine the output tray rather than the page setup option. This option is useful for printing PDFs that contain multiple page sizes on printers that have different-sized output trays.
  - **Scale**: Chose from one of the following scaling options:
    - **None**: Select to not scale any page, whether it fits or not.
    - **Expand to printable area**: Select to expand any page to fit the page area. Pages larger than the paper size are not re-sized.
    - **Shrink to printable area**: Select to shrink any page to fit the page area. Pages smaller than the paper size are not re-sized.

Production Options

- **Booklet Imposition**: Check to tell the printer to generate a booklet for the print output. Booklet options are set in the "Booklet Options" on page 773 page.
  This option is unselected by default unless selected in the Designer "Print section properties" on page 542.
- **Imposition**: Check to enable Cut & Stack Imposition, which is set in the "Imposition options" on page 774 page.
- **Add Inserter marks**: Check to enable Inserter mark functionality, which is set in the "Inserter options" on page 783 page.
- **Runtime Parameters**: select "Runtime Parameter Options" on page 686 to store information which can be used at runtime for comparisons against conditions within Job Creation, or for use with external sorting programs.
- **Override Finishing options** (not available in ***Output Creation Preset*** Wizard): Check to configure custom "Finishing options" on page 691, such as binding.
- **Print virtual stationery**: Check to enable virtual stationery in the output.

- **Use grouping** (not available in ***Output Creation Preset*** Wizard): Check to configure grouping of output into jobs, job segments or document sets. See "Grouping options" on page 696.
- **Include meta data** (not available in ***Output Preset Creation*** Wizard): Check to add meta data to the output. This can be done at Job, Job Segment, Document, Document Set and Page level. See "Meta Data options" on page 699.
- **Separation**: Check to activate the "Separation options" on page 807 page of the wizard.
- **Add additional content**: Check to activate the "Additional Content" on page 732 page of the wizard.

Records

(Not available in ***Output Creation Preset*** Wizard.)

- **Record Range**: Allows selection of a range of records or a custom selection.
  You can specify individual records separated by semi-colons (;) or ranges using dashes.
  For example: 2;4;6-10 would print pages 2, 4, 6, 7, 8, 9 and 10.
- **Apply filtering and sorting to record selection**: Check to activate the "Data filtering options" on page 689 (to filter out certain records) and "Sorting options" on page 693 (to sort the remaining records) pages of the wizard.

Copies

(Not available in ***Output Creation Preset*** Wizard.)

- **Copies**: Enter the number of copies to print, of each record.
- **Collate**: When printing multiple copies you can check this checkbox to have the record copies printed together.
  For example in a three record job the records would print out as 1-1-2-2-3-3, rather than 1-2-3-1-2-3.

Pure Color Thresholds

This section is valid for AFP, IPDS and PCL only. It applies to elements within the record that are shades of grey, rather than black or white.

- **Black Threshold Percentage**: The percentage of shading at which the element will appear as full black, rather than dark grey.
- **White Threshold Percentage**: The percentage at which the element will appear as full white, rather than light grey.

Advanced Print Wizard navigation options

- **Load** button (not available in ***Output Creation Preset*** Wizard): Click to select a previously created Output Creation Preset.
  This will change the Advanced Print Options to match the entries contained within the Preset.

- **Preview** button (active in ***Advanced Print*** Wizard only): Click to launch a Proof Preview window, which displays how the printed output would look based upon the currently chosen selections.
- **Back** and **Next** buttons: Used to navigate back and forth through all the selected options within the Wizard. Up until the Print button is pressed, one can reverse all the way through the wizard to return to the main selection page (the "Print options" on page 704 page) and add or remove printing options from the print run.
- **Print** button (***Advanced Print*** Wizard only) or **Finish** button (***Output Creation Preset*** Wizard only):
  Click to produce print output/finalize the Preset according to the current settings.
  This can be done at any point within the Wizard, whether or not *all* the options selected in the "Print options" on page 704 page have been completed or not.
- **Cancel** button: Cancels the Print Wizard, without creating any printout.

AFP Options

This page appears as part of the **Advanced Print Wizard** and the Output Creation Preset Wizard.

The **AFP Options** page is shown only when a AFP Print output type is selected in the Print Options dialog. It is used to select AFP specific options.

- **Page Groups**
  Select (or deselect) what Page Grouping options are to be created for AFP output.
  - *Job Segments*.
  - *Document Sets*.
  - *Documents*.
- **Options** group:
  AFP specific output options, relating to AFP **T**agged **L**ogical **E**lements (TLEs)

- **Create TLEs for all metadata properties**: Used to control whether automatic TLEs are to be created from job metadata properties and included in the AFP output, or not.

  Prior to Connect 2018.2 TLEs were always created for job metadata properties. This checkbox provides an option for switching off that automatic TLE creation.

  > **Note**
  >
  > If you want to create your own TLEs it can be useful to disable automatic TLE creation. This reduces the chance of unexpected clashes.

- **Rules** selection:
  TLEs and NOPs can be controlled by configurable rules. These rules are organized in groups. Each rule is associated with 1 or more custom AFP records. Those records get inserted into the output when their rule is true.

  - Add. Add a Rule or rule Group. All rules need to reside within rule Groups.

    Groups:
    All rule Groups have a name and trigger.
    The name has no functional purpose other than in identifying the Group within the Table.
    The trigger determines when the rules are evaluated, and they can be set to either *Page*, *Document*, *Document Set* or *Job Segment* triggers.

    Rules:
    Rules can be based upon other groups of rules, or based upon Positioning, Size or Metadata (if Metadata is available).
    Rules within a group act as alternatives, with the first rule equating to "true" being chosen.
  - Edit. Edit a Rule or Group.
    The editing options are the same as those that applied when the Group or Rule was initially created.
  - / Move. The order that groups appear in, controls the order in which the AFP records they control get inserted. Use these controls to sort the order as desired.
  - Delete. Delete the selected Rule(s) or Group(s) from the table.

- **Condition** display box: This displays the contents of the currently selected Rule or rule Group. The contents cannot be edited.
- **Records**:
  Add a TLE or NOP to be applied when the current condition is met. This adds embedded textual metadata in the output, that does not impact upon the appearance of the output in any way.

  TLE (**T**agged **L**ogical **E**lements) records:
  A TLE structured field assigns an attribute name and an attribute value to a page or page group. The TLE structured field may be embedded directly in the page or page group, or it may reference the page or page group from a document index.
  When a TLE structured field is associated with a page group, the parameters of the TLE structured field are inherited by all the pages in the page group and by all other page groups that are nested within that page group.
  The scope of a TLE is determined by its position with respect to other TLEs that reference, or are embedded in, the same page or page group. The TLE structured field does not provide any presentation specifications and therefore has no effect on the appearance of a document when it is created.

  NOP (**N**o **Op**eration) records:
  In some cases, instead of using TLEs, a choice might be made to use NOP records (a field that is bypassed when processing output, and produces no output) with custom data to come up with an entirely proprietary way of adding metadata.

  - Add. Add either a TLE or NOP record to be associated with the current Condition.
    The TLE requires both a name and some textual content.
    The NOP requires just the textual content.
  - Edit. Edit the TLE or NOP entry.
  - / Move. Use these controls to move the selected entry(s) up or down within the Records table.
  - Delete. Delete the selected entries from the table.

Advanced Print Wizard and Output Creation Preset Wizard navigation options

- **Preview** button (active in ***Advanced Print*** Wizard only): Click to launch a Proof Preview window, which displays how the printed output would look based upon the currently chosen selections.

- **Back** and **Next** buttons: Used to navigate back and forth through all the selected options within the Wizard. Up until the Print button is pressed, one can reverse all the way through the wizard to return to the main selection page (the "Print options" on page 704 page) and add or remove printing options from the print run.
- **Print** button (***Advanced Print*** Wizard only) or **Finish** button (***Output Creation Preset*** Wizard only):
  Click to produce print output/finalize the Preset according to the current settings.
  This can be done at any point within the Wizard, whether or not *all* the options selected in the "Print options" on page 704 page have been completed or not.
- **Cancel** button: Cancels the Print Wizard, without creating any printout.

PDF Options

This page appears as part of the **Advanced Print Wizard** and the Output Creation Preset Wizard.

The **PDF Options** page is shown only when a PDF Print output type is selected in the Print Options dialog. It is used to select PDF specific options.

> **Note**
>
> Ideally, PDF should not be used for generating Mixplex output. You should instead use a print format that supports Duplex/Simplex switching. PDF is not suitable for this as it has no device control.

- **PDF Creation Group**
  - **Type**: Use the drop-down list to specify what PDF format will be generated. The options are:
    - Standard PDF (PDF)
    - Archive format PDF (PDF/A-1b)
    - E-invoice format PDF (PDF/A-3b)
    - Graphics format PDF (PDF/X-4)
    - Variable data printing format PDF (PDF/VT).

> **Note**
>
> PDF/A output created from a template that uses CMYK colors will be much bigger than PDF/A output created from a template that uses RGB colors, because PDF/A needs to contain a color profile and the CMYK color profile is rather big compared to a RGB color profile. If output size matters it is recommended to avoid using CMYK colors.

- **Font Creation** selection: The choices are:
  - **Simple fonts when possible (smallest size)**. This is the default option. This option uses Simple fonts in cases where the fonts do not use too many characters. It automatically switches to using composite fonts when the number of characters becomes too large to fit a Simple font. This allows omitting fonts that are considered "standard fonts" in the PDF reference, which helps reduce file size.
    The drawback to this option is that some PDF viewers do not support it properly, which can cause viewing issues with jobs using non-Latin1 encoded characters.

    > **Note**
    >
    > A Simple font can be a Type1 font, a TrueType font or a Type3 font. One of the main characteristics of a Simple font is that its glyphs are selected using a single byte character code, limiting the number of glyphs to a maximum of 256.
    >
    > For more information on Simple fonts, see Abode's documentation regarding Portable Document Format files.

  - **Always create CID fonts (best compatibility)**. This option always uses composite fonts (CID fonts), regardless of the number of glyphs in the font. CID fonts use 2 byte ids to select glyphs, so this option always leads to larger file sizes.
    The benefit of this option is that all the fonts get embedded in the output,

making for the most portable of PDF outputs.
Customers using Asian fonts should generally use the CID font option.

> **Note**
>
> Any pre Connect 2018.2 Output Presets will use the **Simple Fonts** default, unless updated in Connect 2018.2 or later.

- **Embed standard fonts**: Click to embed the 14 standard system fonts within the PDF output. This increases the output filesize but makes the PDF output truly portable. Such PDFs print as displayed on screen, regardless of whether the 14 standard fonts are present on the target printing system or not.

  > **Note**
  >
  > This box is automatically selected for all but standard PDF outputs, as fonts are always embedded in such output types.

- **Pass-through PDF resources**: Click to have PDF resources used *as-is*, without any additional processing. This guarantees the fidelity of any PDF graphics used within the template will be retained in the output.

  This option is only available for standard PDF output.

  > **Note**
  >
  > Connect tries to write content in the best way possible, depending on the chosen output format and optimization settings. Selecting *PDF pass-through* means the output will be less optimized, which typically produces somewhat larger files.

> **Note**
>
> Encrypted PDF files are **not supported** in *PDF pass-through* mode.

- **Keep attachments**: Click to keep any PDF attachments, without any modification. This allows processing of PDF input files with attachments, producing either standard or digitally signed PDFs with attachments.

  This option is not available for PDF/A-1b output.

Metadata
Use this Tab to add optional Metadata to the PDF output. The fields that can be configured are the Title, Author, Description and Keywords. These metadata entries are those that are displayed in the Adobe Reader *Document Properties* dialog. The values for all the fields can be either static or dynamic (such as values obtained from data mapping) or a combination of both.
• **Title**: Add the document name(s). If no entry is made, this defaults to the existing document name (*${template.base}*). • **Author**: Add the document author, if desired. This can be either a static or a dynamic value, or a combination thereof. • **Description**: Add an optional description about the document. A static value is most likely for this field, but it can contain dynamic value(s) as well. • **Keywords**: Optionally add any Keywords that might be beneficial to include. These can be either a static or dynamic values, or a combination thereof. • **Keep PDF/A-3 extension schema** checkbox (option only available for PDF/A-3b output): Click to retain any PDF/A-3 extension schema metadata, if the original PDF input file contained such.

Initial View

Use this Tab to select how the PDF output file is to be displayed upon initial viewing.

The options provided are largely the same as those seen in Adobe Acrobat.

- **Magnification and Layout** selection:
  - **Navigation Tab**: Select what panels should open in the Reader application when viewing the PDF.
  - **Page Layout**: Select how many pages should be displayed upon PDF opening. The choices are:
    - *Default* = Single Page.
    - *Single Page* = Starts with a display sized to fit one entire page, and scrolling down thereafter displays only entire page(s).
    - *Single Page Continuous* = starts with a display of one page. Scrolling down thereafter scrolls onto the next page, with both the first and second page in display.
    - *Display Pages in Two Columns, Left* = the pages appear in two columns, with the odd-numbered pages on the left.
    - *Display Pages in Two Columns, Right* = the pages appear in two columns, with the odd-numbered pages on the right.
    - *Two Pages, Odd Pages Left* = display the pages side by side, with odd-numbered pages on the left .
    - *Two Pages, Odd Pages Right* = display the pages side by side, with odd-numbered pages on the right.
  - **Page Magnification**: Select the level of magnification. The choices are the same as those presented in Acrobat Viewer.
  - **Open to Page**: Select which page is to be the first of the pages to be displayed.
- **Window Options**: Select which of the following application window options to apply:

Initial View

- **Resize window to initial page**: This has the application open to the same size as the initial page.
- **Centre Windows on screen**: Centre the application on screen.
- **Open in full screen mode**: Open the application in full screen mode.

- **User Interface Options**: Chose which application interface options to hide, if any.
  Select from the following: **Menu bar**; **Tool bars**; **Windows controls**

Digital Signature

A digital signature identifies the person signing a document, similarly to a conventional handwritten signature.

A digital signature is more difficult to forge than a handwritten signature as it contains encrypted information which is unique to the signer and which can be password protected and verifiable.

> **Note**
>
> PrintShop Mail Connect currently supports only a single Digital Signature per PDF output file.

If the PDF has attachments, signing it requires a change in the print process via Workflow. The Workflow process is as follows:

1. Use the PDF as an input in the Execute Data Mapping task and enable **Bypass content creation**.
2. Feed the result of the Execute Data Mapping task to the Create Job task. Omit the Create Print Content task.

Digital Signature

3. Next up is the Create Output task which will then sign the PDF while preserving any attachments.

- **Digital signature**: Check the box to enable the integration of a digital signature within the PDF

  The Digital signature table contains the Digital signature Keystore and associated Signature.
  The table has the following columns:

  - **Name**: The user-defined name of the Keystore\Signature.
  - **Alias**: The user-defined alias for the signature.

  The table allows the following options:

  - **Add Keystore**: opens the "Keystore" on page 721 dialog for adding a Digital signature Keystore.
    Digital signatures *require* a Keystore, so initially only the **Add Keystore** option will be available.
  - **Add Signature** button only becomes active once a Keystore has been added. Selecting this option opens the PDF Signature dialog for adding a new PDF signature to the Keystore.
  - **Edit**: Click to edit the currently selected item. Either the Keystore or PDF Signature.
  - **Delete**: Click to delete the currently selected item. Deleting the Keystore will also delete the associated Signature.

Optimize

Use the options in this tab to optimize the PDF output.

- **Optimize Images** selection: Select to enable optimization of images, with different settings available for Color Images, Grey Images and Monochrome

Optimize

Images.
Each of the image types share the same basic options, with individual options tailored to the selected Image type also available.
The options are:

- **Compression**: Select what compression to use, if any.
  The default option is to *Retain Existing* settings, which disallows any further alteration of compression for this Image type.
- **Download Filter**: This choice becomes available only when a Compression type other than *Retain Existing* is selected.
  Select which Download Filter to use, if one is to be used. The default is to not include one (*Off*).

  If a download Filter has been selected, then set the desired display resolution in Pixels Per Inch (PPI). Select the maximum resolution (the **when Images PPI is above** value), and what resolution is to be used if the selected maximum is exceeded (the **set PPI to** value).

- **Optimize for fast web view**: Check this box to allow linearizing the PDF output. This allows readers of the PDF to start reading the PDF online before the entire PDF downloads.

Security

Use the options in this tab to set the PDF output security. The options mirror those available in Adobe Acrobat.

- **Open Document** checkbox: Select this option to add password protection to the output PDF, which allows the document be opened only after entering the same **Password** as that selected here.
  To view the password entry in plain text, or to toggle back to masked view, click the Password button.
  - **Password is dynamic** checkbox. Select this if you wish to use a variable

Security

(such as a metadata field or a data file field) for the password.
The datafile selection format is ${datafield}. For example, *${document.metadata.password}*

- **Permissions**: checkbox: Select this option to set optional Permissions. These include a choice of the following:
  - **Allow Printing**: select whether printing is to be disallowed (*None*), or whether to restrict to low quality printing only (*Low Quality*), or whether to allow full printing (*High Quality*).
  - **Changing the document**: whether the document itself is editable, with changes allowed.
  - **Content copying**: whether the contents of the document are allowed to be copy and pasted or not.
  - **Annotations and comments**: whether the document can be commented upon.
  - **Document assembly**: whether the Document assembly is available.
  - **Form filling**: whether PDF forms elements are allowed to be filled.
  - **Screen readers (Accessibility)**:whether the PDF will be WCAG capable and allow screen readers to convert the text into spoken word.
  - **Permission password**: Select a **Password** that allows the Permissions to be edited in the output document.
    To view the password entry in plain text, or to toggle back to masked view, click the Password button.
    - **Password is dynamic** checkbox. Select this if you wish to use a variable (such as a metadata field or a data file field) for the password.
      The datafile selection format is ${datafield}. For example, *${document.metadata.password}*

Advanced Print Wizard and Output Creation Preset Wizard navigation options

- **Preview** button (active in ***Advanced Print*** Wizard only): Click to launch a Proof Preview window, which displays how the printed output would look based upon the currently

chosen selections.

- **Back** and **Next** buttons: Used to navigate back and forth through all the selected options within the Wizard. Up until the Print button is pressed, one can reverse all the way through the wizard to return to the main selection page (the "Print options" on page 704 page) and add or remove printing options from the print run.
- **Print** button (***Advanced Print*** Wizard only) or **Finish** button (***Output Creation Preset*** Wizard only):
  Click to produce print output/finalize the Preset according to the current settings.
  This can be done at any point within the Wizard, whether or not *all* the options selected in the "Print options" on page 704 page have been completed or not.
- **Cancel** button: Cancels the Print Wizard, without creating any printout.

Keystore

The security certificate **Keystore** dialog appears when adding or editing a keystore from the "PDF Options" on page 712 page. This dialog allows you to select a keystore with a private key.

The keystores currently supported by Connect are:

- JKS (Java Key Store) format.
- JCEKS (Java Cryptography Extension Key Store) format.
- PKCS#12
- PKCS#11

> **Note**
>
> *JKS* and *JCEKS* represent file based keystores in a Java proprietary format. The Java keytool command line application can be used to create and manage keystores in this format.
>
> Type *PKCS#12* also represents a file based keystore, but in the PKCS#12 format.
>
> Type *PKCS#11* represents a keystore on a hardware device. These hardware devices must first be setup in the "Hardware for Digital Signing preferences" on page 385 **Preferences** page, before they become available as an option in this dialog.

These are the options available in this dialog:

- **Name**: Enter a name for the keystore to describe it within Connect.
- **File**: Enter the path to the keystore file, or use the Browse button to locate the file. This option is not relevant for *PKCS#11* hardware devices.
- **Keystore** properties group:
  - **Type**: Use the drop-down to select the appropriate type of keystore format.
  - **Password**: If the keystore is password protected, type in the password that secures the keystore.
  - **Repeat Password**: Re-type in the password that secures the keystore. Once this is done the two Password entry boxes will no longer have the red cross icon (indicating incomplete or unselected) flag beside them.
- **PKCS#11** properties group:
  - **Module Name**: Select which secure hardware device (USB tokens, smart cards, and Hardware Security Modules) to connect to through PKCS#11. These hardware devices must first be setup in the "Hardware for Digital Signing preferences" on page 385 **Preferences** page, before they become available as an option in this dialog.
  - **Slot list index**: Select which hardware slot to use.

    > **Note**
    >
    > Slots are logical partitions in the hardware device. In case of Hardware Security Modules, there could be hundreds or more slots are available while in the case of smart cards, there would likely only be one slot available.
    >
    > Specifying both a *Slot list index* and a *Slot ID* is not allowed. Enter one or the other, or neither.

  - **Slot ID**: Select which hardware Slot ID to use.

    > **Note**
    >
    > Slots are logical partitions in the hardware device. In case of Hardware Security Modules, there could be hundreds or more slots are available while

in the case of smart cards, there would likely only be one slot available.

Specifying both a *Slot list index* and a *Slot ID* is not allowed. Enter one or the other, or neither.

- **Properties file group**:
  - **File**: Load optional keystore properties file. Properties files could be used for storing the password, or similar.

PDF Signature

The **PDF Signature** dialog appears when either adding or editing a Signature on the "PDF Options" on page 712 page.

The PDF digital signature contains information about the signer, the location and reason used for signing and the time of creation.

The digital signature can be made visible on the output, with the dimensions of the signature box being configurable.

Note

PrintShop Mail Connect currently supports only a single Digital Signature per PDF output file.

- **Name**: Enter a name that describes the signature entry.
- **Keystore name**: Use the drop-down to select which keystore the signature is pulled from. These keystores are set in the "Keystore" on page 721 dialog, called from the "PDF Options" on page 712 page.
- **Signature Properties group**: These are optional Metadata fields associated with the signature, which can be omitted.
  - **Location**: The CPU host name or physical location of the signing.
  - **Reason**: Records the reason for the signing.
  - **Contact**: Information to enable a recipient to contact the signer to verify the signature. For example: a phone number.

- **Key group**: Refers to a key from the keystore.
  - **Alias**: The user-friendly name of the key.
    To select the alias directly from those present in the specified keystore, press the **Chose Alias from keystore** button ( ) to the right of the edit box. This launches the "Chose Alias from Keystore" on the next page dialog.
  - **Password**: Enter the password for the key (the same password as was entered in Key Store).
  - **Repeat Password**: Re-enter the password for the key (same as previous).
- **Timestamp**: Check to enable time stamping authentication.

> **Note**
>
> Not available for signatures set to use Adobe.PPKLite Handler.

  - **Timestamp Server URL**: Select the Time Stamp Authority (TSA) URL address. One example of a TSA is https://freetsa.org.
  - **Account**: Account name specific to the TSA server chosen.
  - **Password**: Password specific to the TSA server chosen.
  - **Repeat Password**: Repeat of password.
- **Visible Signature**: Check to add a visible signature to the PDF file.

  The signature is added to the specified page of each output file.

  The signature will be created using Arial 10pt black font. If this font does not exist on the production machine, a substitute font will be used.

  If the digital signature contents exceed the dimensions of the configured box, the text will be scaled down.

> **Note**
>
> When creating PDF/VT-1 or PDF/X-4 output, the visible signature should be located outside the visible page area (trimbox/bleedbox).

The Weaver engine logs a warning message if this is not the case.

The default measuring option is points (pt) but metric options (mm / cm) can be manually selected.

- **X**: Enter the horizontal distance between the left side of the page and the left side of the signature.
- **Y**: Enter the vertical distance between the top of the page and the top of the signature.
- **Width**: Enter the desired width of the signature.
- **Height**: Enter the desired height of the signature.

Select which **Page** the signature is to be embedded in:

- **First Page**
- **Last Page**
- **Specific Page**. Enter which page in the **Page number** edit field, which becomes enabled when this option is selected.
  If the output file does not contain the specified page, then the signature will be added as an invisible signature to the PDF output file.

Advanced Print Wizard navigation options

- **Preview** button: Click to launch a Proof Preview window, which displays how the printed output would look based upon the currently chosen selections.
- **Back** and **Next** buttons: Used to navigate back and forth through all the selected options within the Wizard. Up until the Print button is pressed, one can reverse all the way through the wizard to return to the main selection page (the "Print options" on page 704 page) and add or remove printing options from the print run.
- **Print** button: Click to produce print output according to the current settings. This can be done at any point within the Wizard, whether or not *all* the options selected in the "Print options" on page 704 page have been completed or not.
- **Cancel** button: Cancels the Print Wizard, without creating any printout.

Chose Alias from Keystore

Select from which alias to use, from the list of aliases for keys that can be found in the keystore. If any such aliases exist and can be read.

If the keystore cannot be accessed to retrieve those aliases, a message will be shown stating that the keystore cannot be accessed.

If the keystore does not contain any aliases or if they cannot be read, then a message will be shown stating that aliases could not be obtained.

PPML Options

This page appears as part of the **Advanced Print Wizard** and the Output Creation Preset Wizard.

The **PPML Options** page is shown only when a PPML Print output type is selected in the Print Options dialog. It is used for tray selection.

PPML Tray Options

Use the drop-down list to specify what output Tray Options to use, if any. The options are as follows:

- **None**: No Tray Options specified. This is the default selection.
- **SetPageDevice**: This provides a table where one can allocate job Media to different types of print paper stock and assign them to the output via the SetPageDevice entry. The choices are a selection of common print stock Types and Colors.
  - **Media** column: The available job Media . These will initially be populated with the Job Media entries, but new Media can also be added via the Add button.
  - **Type** column: Select the Media Type from the range available. The selections are standard print output media types, except for the Unspecified option, which resolves to the printer default, or to that which matches the PageSize setting.
  - **Color** column: Select the Media color from the range available, if needed.

  The SetPageDevice table options are:
  - **Add**: Add new Media entries.
  - **Delete**: Delete the selected Media from the table.
  - / **Move Up / Down**: Move the selected Media up or down within the table.
  - **Import**: Import additional Media entries from existing templates.

    Choose from:
    - **Import from current template**: As the Media selections from the current template are automatically added to the table, selecting this option would

either duplicate the existing entries (if they had not been deleted) or would repopulate the table with those entries (had they been deleted).
If the existing entries had not been deleted, then the newly imported Media names will be identical to the entries already in the table, and thus their names will need to be changed.

- **Import from template file**: Import Media settings from some other Connect Template file.

- **Mapped PPD Entries**: This allows selection of a printer manufacturer's printer specific PostScript Printer Definition (PPD) file. These files contain what printing options are available for this specific printer.
  - **PPD Info** box: This is a read only box that contains the name of the selected PPD file, once such a selection is made.
  - **PPD file name**: Select a PostScript Printer Definition (PPD) file, using the Browse button.

    Once the PPD is loaded, you will be asked to save this new Printer Definition type. When the new Printer Definition is saved, the PPD Info box will be populated with information regarding it, and the following new dialog options become available:
    - **Media**: This box will initially be populated with any existing Media found in the current Template. You can add or remove Media, as you wish. The options available to the Media table are:
      - **Add**: Add new Media entries. This launches the "Dynamic Tray Mapping Editor Dialog" on page 800 to allow you to set the Media properties.
      - **Delete**: Delete the selected Media from the table.
      - / **Move Up / Down**: Move the selected Media up or down within the table.
      - **Import tray settings**: Import additional Media entries from existing templates. The choices are:
        - **Import from current template**: As the Media selections from the current template are automatically added to the table, selecting this option would either duplicate the existing entries (if they had not been deleted) or would repopulate the table with those entries (had they been deleted).
          If the existing entries had not been deleted, then the newly imported Media names will be identical to the entries already in the

table, and thus their names will need to be changed.

- **Import from template file**: Import Media settings from some other Connect Template file.

Double clicking the left mouse button on a Media table entry will launch the "Dynamic Tray Mapping Editor Dialog" on page 800 to allow you to modify that Media's properties.

- **Details**: This is a read only table that displays the Details of whatever Media has been selected in the Media box. Double clicking the left mouse button on a Media table entry will launch the "Dynamic Tray Mapping Editor Dialog" on page 800 to allow you to modify that Media's properties.
- **Printer Options**:
  The **Printer Options** table displays :
  - **Name** column: This column contains the names of the Printer Options, as taken from the PPD file.
  - **Option** column: This column shows the selected Printer Option preference.
  - **Order** column: This column displays any Dependency Order criteria that applies to the selected Printer Option.

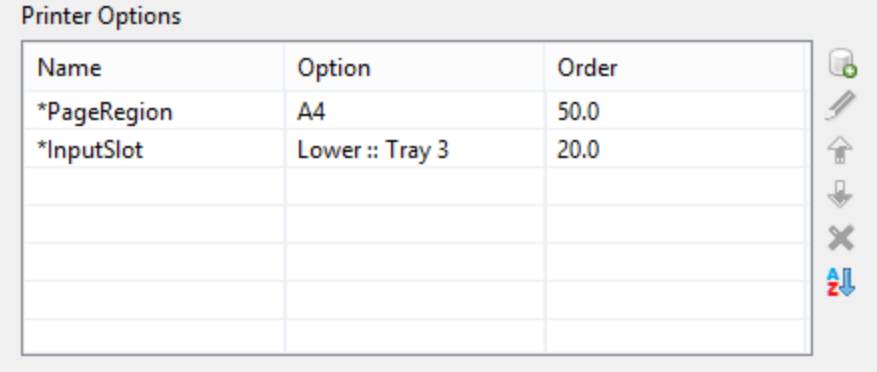
  The choices available to the Printer Options are as follows:
  - **Add**: This launches the "Add Printer Option dialog" on page 802 which allows you to add one or more Printer Options.
  - **Edit**: This allows you to edit an existing Rule Printer Option. It launches the "Add Printer Option dialog" on page 802.
  - / **Move Up / Down**: Move the selected Printer Option(s) up or down within the table. This allows sorting of options for easier legibility, but manually moving options up and down is done without any check on Order Dependency.
  - **Delete**: Delete the selected Printer Option(s).
  - **Sort by Order dependency**: This sorts the Printer Options by the order of their dependencies.
    Not all Printer Options have dependencies, but for those that do, the order is important.
    For example, consider these following PPD file Printer Option extracts:
    - ***PageRegion:***
      `*OpenUI *PageRegion: PickOne`

```
*OrderDependency: 50 AnySetup *PageRegion
...
*CloseUI: *PageRegion
```

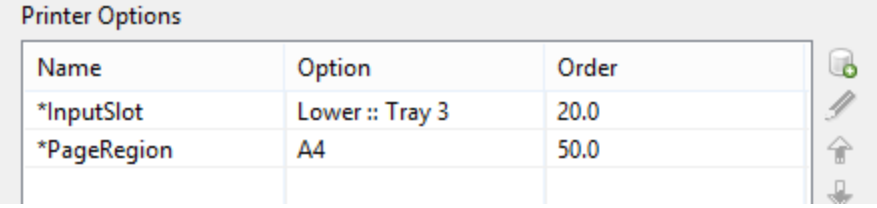
- ***InputSlot:***

```
*OpenUI *InputSlot: PickOne
*OrderDependency: 20 AnySetup *InputSlot
...
*CloseUI: *InputSlot
```

We can see that one has an Order Dependency of 50 and the other 20. If *PageRegion* was selected prior to *InputSlot* in the "Add Printer Option dialog" on page 802, then they would appear like this in the Print Wizard:



To sort them into the correct dependance order, use the **Sort by Order dependency** button. This will sort them into the proper dependency sequence. As seen below, where the lower dependency now appears first.



Warning

Nothing prevents the selection of conflicting and contradictory Print Options.

This can be seen in the following example, which shows multiple conflicting Folding Options have been chosen:

Printer Options

Name	Option	Order
*Punch :: Punch	None :: Off	1.0
*Staple :: Staple	None :: Off	21.0
*Prepunched :: Pre-Punched	True :: On	65.0
*SquareFold :: Spine Corner Forming	True :: On	1.0
*MultiFolder :: Multi Folder	FD503(2-4H) :: Multi F...	1.0
*Fold :: Fold	ZFold :: Z-Fold	1.0

The breadth and diversity of PPD files as well as individual printer capabilities mean that the Print Wizard cannot meaningfully validate the selections made, and thus does not attempt to.

So you must be careful when making selections. We recommend that you always test each individual selection made, to confirm that they are valid and that they do what you expected of them.

This could entail considerable trial and error, but should lead to less anguish in the long run.

Advanced Print Wizard and Output Creation Preset Wizard navigation options

- **Preview** button (active in ***Advanced Print*** Wizard only): Click to launch a Proof Preview window, which displays how the printed output would look based upon the currently chosen selections.
- **Back** and **Next** buttons: Used to navigate back and forth through all the selected options within the Wizard. Up until the Print button is pressed, one can reverse all the way through the wizard to return to the main selection page (the "Print options" on page 704 page) and add or remove printing options from the print run.

- **Print** button (***Advanced Print*** Wizard only) or **Finish** button (***Output Creation Preset*** Wizard only):
  Click to produce print output/finalize the Preset according to the current settings.
  This can be done at any point within the Wizard, whether or not *all* the options selected in the "Print options" on page 704 page have been completed or not.
- **Cancel** button: Cancels the Print Wizard, without creating any printout.

Spot Color Settings

The **Spot Color Settings** page is used to match Spot Colors contained within a job to the color settings available on the printer.
This option is only available for AFP and IPDS print outputs, and requires a Printer Definition File (see "Print preferences" on page 390) that supports Spot Colors.

- **SpotColor Mapping** table:
  - **SpotColor Name** column: This contains the names of the Spot Colors used within the job.
    If the Spot Color were added to the job within Connect Designer, then those Spot Colors will be automatically included in the table.
  - **SpotColor Number** column: This lists which Spot Color available on the printer is to be associated with the named Spot Color.
- **Add**: Add a new association between a Spot Color in the job, and the Spot Colors options available on the printer.

> **Note**
>
> If the Spot Color was not added to the job within Connect Designer itself but are instead included within a external file (such as a PDF) added into the job, then care must be taken to ensure the Name added here is identical to the name reference used within that included file.

- **Delete**: Delete a Spot Color association.
- **Edit**: Edit an existing Spot Color association.

Additional Content

This page appears as part of the **Advanced Print Wizard** and the Output Creation Preset Wizard.

The **Additional Content** page allows you to add content at the time of printing. There are four different types of Additional Content that can be added at print time: **Text**; **Images**; **Barcodes** and **OMR Marks**.
They can be used to add either static content or variable content.

The Additional Content option is particularly useful when you might need to drive custom processes on production machines using either Barcodes or OMR Marks, or if you need to add some last minute additions to the print job via text and/or images.

Page breakdown

The Additional Content table displays any Additional Content that has been set.
Additional Content can be added, edited, removed, duplicated or moved within the table via the buttons to the right of the table.

The **Additional Content options** are:

- **Add**: Click to open some Additional Content.
  The choices are as follows:
  - "Additional Text Settings" on page 734 entry.
  - "Additional Barcode Options" on page 737 entry, from the selection presented.
  - "Additional Image Settings" on page 735 entry.
  - "Additional OMR Mark Settings" on page 768 entry.
- **Edit**: Click to edit the currently selected entry. This will launch the appropriate edit box, for the selected entry type.
  Double clicking on an entry in the table has the same effect as this button.
- **Delete**: Click to delete the currently selected entry or entries.
- **Duplicate**: Click to create a duplicated copy of the entry. This creates a new entry that is exactly the same as the original.
- / **Move Up / Move Down**: Click to move the selected entry (or entries) up or down within the table.

Additional Content Table: The table is split up into the following columns

- **Include in Output**: Indicates whether the entry is to be included in the output or not.
  The settings made in the Additional Content dialog box can be over-ridden here.
  This allows you to create a virtual "library" of Additional Content options in an Output Preset, from which you can then pick and choose from at time of printing.
- **Type**: Displays an icon showing what type the Additional Content entry is.
- **Description**: Displays the Description" that was entered in the Additional Content dialog.

> **Note**
>
> The entered text Description might stretch over a couple of lines, so it is possible that not all of the Description will be displayed here.

- **Left**: Displays the Additional Content Left Positional value.
- **Bottom**: Displays the Additional Content Bottom Positional value.
- **Content**: Displays the actual Additional Content that is to be inserted.
  In the case of Text, this would be the actual text (or data fields) that are to be inserted.
  In the case of Image, this would be the name of the image file.
  In the case of Barcode, this would be the data fields used for creating the barcode.
  In the case of OMR, this would be the number of marks set for this entry.
- **Condition**: Displays the Conditional entry that is used to determine whether this Additional Content instance is to be added to the output or not.

Advanced Print Wizard and Output Creation Preset Wizard navigation options

- **Preview** button (active in ***Advanced Print*** Wizard only): Click to launch a Proof Preview window, which displays how the printed output would look based upon the currently chosen selections.
- **Back** and **Next** buttons: Used to navigate back and forth through all the selected options within the Wizard. Up until the Print button is pressed, one can reverse all the way through the wizard to return to the main selection page (the "Print options" on page 704 page) and add or remove printing options from the print run.
- **Print** button (***Advanced Print*** Wizard only) or **Finish** button (***Output Creation Preset*** Wizard only):
  Click to produce print output/finalize the Preset according to the current settings.
  This can be done at any point within the Wizard, whether or not *all* the options selected in

the "Print options" on page 704 page have been completed or not.

- **Cancel** button: Cancels the Print Wizard, without creating any printout.

Additional Text Settings

The **Additional Text Settings** dialog displays the property of Text added in the "Additional Content" on page 732 page.

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.
  - **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.
- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the Text added to the page.
  - **Output once per sheet**: Option relates to Imposition (also known as N-Up) printing. Select this box to have the Text printed once per sheet rather than once per document page.

    > **Note**
    >
    > If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

  - **Left**: Enter the distance between the left margin of the page and the Text, in either metric (cm/mm), inch (in), pixel (px) or point (pt) values.
  - **Bottom**: Enter the distance between the bottom margin of the page and the Text, in either metric (cm/mm), inch (in), pixel (px) or point (pt) values.
- **Font** group:

- **Font Name**: Use the drop-down to select which font type to apply to the Text. The drop-down displays all the fonts installed on the system.

  > **Note**
  >
  > Vertically-oriented fonts (fonts whose typeface name begin with the at (@) character) are not supported in Connect.

  - **Style**: Choose between "Regular", "*Italic*", "**Bold**" or "***Bold Italic***".
  - **Font Size**: Enter the font size in points (pt).
  - **Color**: Select what color the Text will be.
- **Text**: Enter the actual Text to appear on the page in the selected location. The Text can be spread over multiple lines, but no additional formatting can be added within this edit box. The entire Text will be printed using the formatting options selected in the **Font group**.

  Use the **Add** button to display a list of variable data that can be added to the Text.

  This includes metadata fields added in the Metadata Options, as well as some document information fields (see "Print output variables" on page 1008 for information on those).
- **Condition**: Enter the condition which determines whether or not the Text will be added to the document at print time.

  Use the **Add** button for selection options.

  For details on how to create a conditional, see the "How to Set Conditionals" on page 772 page.

Additional Image Settings

The **Additional Image** dialog displays the properties of the image added in the "Additional Content" on page 732 page.

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets,

libraries from which you can pick and choose what entries you wish to have included, at time of printing.

- **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.

- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the image.
  - **Layer**: Whether this image will appear behind the text (the text will print over the image) or in front of the text (the text behind will be blanked out by the image, as transparent images are not supported).
    - **Ouput once per sheet**: Option relates to Imposition (also known as N-Up) printing. Select this box to have the Image printed once per sheet rather than once per document page.

> **Note**
>
> If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

  - **Left**: Enter the distance between the left margin of the page and the image, in either metric (cm/mm), inch (in), pixel (px) or point (pt) values.
  - **Bottom**: Enter the distance between the bottom margin of the page and the image, in either metric (cm/mm), inch (in), pixel (px) or point (pt) values.
- **Filename**: Use the browse button to launch a Browse box to select an image file. This is a mandatory field.

> **Note**
>
> Transparent images are not supported.

- **Preview** group: This displays a preview of the selected Image.
- **Scaling** group:
  Scaling the image expands the image but keeps the aspect ratio. The amount of scale

and specific limitations can be applied used a combination of the following options:

- **Max Width**: Enter the absolute maximum width the image can be scaled to, in either metric (cm/mm), inch (in), pixel (px) or point (pt) values.
- **Max Height**: Enter the absolute maximum height the image can be scaled to, in either metric (cm/mm), inch (in), pixel (px) or point (pt) values.
- **Scale**: What scale to apply to the image. The maximum scale is 10.0 to 1. Decimal values are allowed for this field.

- **Condition**: Enter the condition which determines whether or not the image will be added to the document at print time. Use the button for selection options.

  For details on how to create a conditional, see "How to Set Conditionals" on page 772 page.

Additional Barcode Options

When adding Barcodes in the "Additional Content" on page 732 page you can select from a series of predetermined Barcode types.
To create dynamic barcodes, "Meta Data options" on page 699 must first have been set. Metadata fields are required to create the association between the dynamic data used in the print run and the barcode.

The options for each of these types are described on the following pages:

- "Australia Post Settings" on page 631
- "Codabar Settings" on the facing page
- "Code 39 Settings" on page 743
- "Code 128 Settings" on page 740
- "Data Matrix settings" on page 745
- "EAN-8 Settings" on page 753
- "EAN-13 Settings" on page 751
- "GS1-128 Settings" on page 748
- "Interleaved 2 of 5 Settings" on page 755
- "Japan Post Settings" on page 652
- "KIX Code (Dutch Post) Settings" on page 654

- "PDF417 Settings" on page 757
- "QR Code Settings" on page 760
- "Royal Mail 2D Mailmark Settings" on page 662
- "Royal Mail 4 State (CBC) Settings" on page 664
- "Royal Mail 4 State Mailmark C Settings" on page 666
- "Royal Mail 4 State Mailmark L Settings" on page 668
- "UPC-A Settings" on page 763
- "UPC-E Settings" on page 766
- "US Postal Service IMb Settings" on page 675
- "US Postal Service IMpb Settings" on page 677

Codabar Settings

Codabar barcodes support the following data: 0-9 - $: / . + plus the optional specification of start/stop characters.

> **Note**
>
> To create dynamic barcodes, Metadata options must first have been set (see "Meta Data options" on page 699).
> Metadata fields are required to create the association between the dynamic data used in the print run and the barcode.

Use the following options to configure the output Barcode settings:

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.
  - **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.
- **Position** group:

- **Orientation**: Use the drop-down to select the orientation of the Barcode added to the page.
- **Output once per sheet**: This option relates to Imposition printing (see Imposition), also known as N-Up printing. Select this box to have the Barcode printed once per sheet rather than once per document page.

> **Note**
>
> If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

- **Left**: Enter the distance between the left margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
- **Bottom**: Enter the distance between the bottom margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.

- **Codabar Properties** group:
  - **Height**: Enter the Barcode height in either metric (cm/mm), inch (in) or point (pt) values.
  - **Module Width**: Specifies the width of the narrow bars. Changing this value to a higher value will generally make the Barcode bigger.
  - **Bar width ratio**: Set the Barcode bar width.
  - **Default start symbol**: Use the drop-down to select the optional Barcode start character, which defines the encoding mode.
  - **Default stop symbol**: Use the drop-down to select the Barcode stop character, which defines the encoding mode.
    - **Print human readable text:** Check to add a textual version of the Barcode data.
    - **Placement**: Use the drop-down to select whether to place the human readable text above or below the Barcode.
    - **Font name**: Use the drop-down to select the font with which to display the human readable text.

> **Note**
>
> Vertically-oriented fonts (fonts whose typeface name begin with the at (@) character) are not supported in Connect.

  - **Font size**: Enter a font size for the human readable text.
  - **Display start/stop symbols** checkbox: Adds the stop/start symbols to the Barcode text.

- **Process Tilde**:
  This selection is not applicable to this barcode type.

- **Text**: Enter the text used to generate the Barcode.
  - **Add** button: Click to display a list of variable data that could be used for generating the Barcode.
    This includes metadata fields which must previously have been added in the Metadata options, likely at the *Document Tags* level (see Metadata Options), as well as some information fields.
- **Condition**: Enter the condition which determines whether or not the Barcode will be added to the document at print time. For details on how to create a conditional, see "How to Set Conditionals" on page 772.
  - **Add** button: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

Code 128 Settings

Code 128 is a high-density barcode, used for alphanumeric or numeric-only barcodes. It supports all 128 ASCII characters.

> **Note**
>
> To create dynamic barcodes, Metadata options must first have been set (see "Meta Data options" on page 699).
> Metadata fields are required to create the association between the dynamic data used in

the print run and the barcode.

Use the following options to configure the output Barcode settings:

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.
  - **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.
- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the Barcode added to the page.
  - **Output once per sheet**: This option relates to Imposition printing (see Imposition), also known as N-Up printing. Select this box to have the Barcode printed once per sheet rather than once per document page.

    Note

    If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

  - **Left**: Enter the distance between the left margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
  - **Bottom**: Enter the distance between the bottom margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
- **Code 128 Properties** group:
- **Height**: Enter the Barcode height in either metric (cm/mm), inch (in) or point (pt) values.

- **Module Width**: Specifies the width of the narrow bars. Changing this value to a higher value will generally make the Barcode bigger.
  - **Print human readable text:** Check to add a textual version of the Barcode data.
  - **Placement**: Use the drop-down to select whether to place the human readable text above or below the Barcode.
  - **Font name**: Use the drop-down to select the font with which to display the human readable text.

    > **Note**
    >
    > Vertically-oriented fonts (fonts whose typeface name begin with the at (@) character) are not supported in Connect.

  - **Font size**: Enter a font size for the human readable text.
- **Process Tilde**:
  Check this option to process tilde characters in the data as special characters.

  Note: with this option checked, any tilde that needs to be included in the output must be escaped by adding another tilde: ~~.
- **Text**: Enter the text used to generate the Barcode.
  - **Add** button: Click to display a list of variable data that could be used for generating the Barcode.
    This includes metadata fields which must previously have been added in the Metadata options, likely at the *Document Tags* level (see Metadata Options), as well as some information fields.
- **Condition**: Enter the condition which determines whether or not the Barcode will be added to the document at print time. For details on how to create a conditional, see "How to Set Conditionals" on page 772.
  - **Add** button: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

Code 39 Settings

Code 39 is a discrete, self-checking barcode that is also known as "Alpha39", "Code 3 of 9" (often abbreviated to "3 of 9"), "Code 3/9", "Type 39", "USS Code 39" and "USD-3".

Code 39 data should contain no more than 20 digits from within the following range: Numeric digits: (0-9), upper-case letters (A-Z), seven special characters (- . space $ / + %) and the start/stop asterisk (*) character.

If the Extended character set is chosen, then lower-case letters (a-z) and other special ASCII characters can also be included.

Note

To create dynamic barcodes, Metadata options must first have been set (see "Meta Data options" on page 699).
Metadata fields are required to create the association between the dynamic data used in the print run and the barcode.

Use the following options to configure the output Barcode settings:

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.
  - **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.
- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the Barcode added to the page.
  - **Output once per sheet**: This option relates to Imposition printing (see Imposition), also known as N-Up printing. Select this box to have the Barcode printed once per sheet rather than once per document page.

> **Note**
>
> If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

  - **Left**: Enter the distance between the left margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
  - **Bottom**: Enter the distance between the bottom margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.

- **Code 39 Properties** group:
  - **Height**: Enter the Barcode height in either metric (cm/mm), inch (in) or point (pt) values.
  - **Use extended character set**: Check to use the Code 39 Extended character set. This extends the range of supported data to include the full ASCII character set. This adds support for lower case letters (a-z) and the full range of ASCII punctuation and special characters.
  - **Module Width**: Specifies the width of the narrow bars. Changing this value to a higher value will generally make the Barcode bigger. The smallest Module Width is 0.19mm (high density).
  - **Bar width ratio**: Set the Barcode bar width.
    - **Checksum**: Use the drop-down to select how to deal with the Barcode checksum:
    - **Ignore**: Ignore checksum calculations.
    - **Auto**: Add a checksum character to the Barcode if the initial value does not validate. This is the default value.
    - **Check**: Verify the Barcode has a valid checksum.
    - **Add**: Calculate and add a checksum character to Barcode, regardless of current value.
    - **Print human readable text:** Check to add a textual version of the Barcode data.
    - **Placement**: Use the drop-down to select whether to place the human readable text above or below the Barcode.

- **Font name**: Use the drop-down to select the font with which to display the human readable text.

  > **Note**
  >
  > Vertically-oriented fonts (fonts whose typeface name begin with the at (@) character) are not supported in Connect.

- **Font size**: Enter a font size for the human readable text.

- **Process Tilde**:
  Check this option to process tilde characters in the data as special characters.

  Note: with this option checked, any tilde that needs to be included in the output must be escaped by adding another tilde: ~~.

- **Text**: Enter the text used to generate the Barcode.
  - **Add** button: Click to display a list of variable data that could be used for generating the Barcode.
    This includes metadata fields which must previously have been added in the Metadata options, likely at the *Document Tags* level (see Metadata Options), as well as some information fields.
- **Condition**: Enter the condition which determines whether or not the Barcode will be added to the document at print time. For details on how to create a conditional, see "How to Set Conditionals" on page 772.
  - **Add** button: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

Data Matrix settings

A Data Matrix barcode is a high-density, two-dimensional (2D) matrix barcode which supports encoded text, numbers, files and digital data.

> **Note**
>
> In order to create a **GS1** compliant Data Matrix barcode, the barcode data must meet the

following requirements:

- The barcode data must start with a leading FNC1 character. Either ~1 or ~d232.
- The GS1 Application Identifiers (AI) must be used for all data.
  The function code ~1 must be used as field separator for variable length AI elements.
- Only ASCII characters should be used.

Use the following options to configure the output Barcode settings:

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.
  - **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.
- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the Barcode added to the page.
  - **Output once per sheet**: This option relates to Imposition printing (see Imposition), also known as N-Up printing. Select this box to have the Barcode printed once per sheet rather than once per document page.

    Note

    If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

  - **Left**: Enter the distance between the left margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.

- **Bottom**: Enter the distance between the bottom margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.

- **Data Matrix Properties** group:
  - **Module Width**: Specifies the width of the narrow bars. Changing this value to a higher value will generally make the Barcode bigger.
  - **Encoding**: The data represented in the symbol can be compressed using one of the following algorithms:
    - **Auto**: Automatically detect the data content and encodes using the most appropriate method. This is the default option.
    - **ASCII**: is used to encode data that mainly contains ASCII alphanumeric characters (ASCII 0-127). Use where Barcode size is a concern and where the data is alphanumeric.
    - **Base 256**: used to encode 8-bit values.
    - **C40**: used for data that mainly consists of numbers and upper-case alphabetic letters.
    - **Text**: used for data that mainly consists of numbers and lower-case alphabetic letters.
    - **None**: Does not use any encoding.
  - **Format**: select the Barcode size format from the drop-down list .

- **Process Tilde**:
  Check this option to process tilde characters in the data as special characters.

  Note: with this option checked, any tilde that needs to be included in the output must be escaped by adding another tilde: ~~.

> **Note**
>
> The following tilde codes are supported in GS1 DataMatrix barcodes:
>
> - ~1 = FNC1 character (for GS1 DataMatrix barcodes)
> - ~2 = Structure Append
> - ~3 = Reader programming
> - ~5 = Macro5

- ~6 = Macro6
- ~7 = ECI expressions

- **Text**: Enter the text used to generate the Barcode.
  - **Add** button: Click to display a list of variable data that could be used for generating the Barcode.
    This includes metadata fields which must previously have been added in the Metadata options, likely at the *Document Tags* level (see Metadata Options), as well as some information fields.
- **Condition**: Enter the condition which determines whether or not the Barcode will be added to the document at print time. For details on how to create a conditional, see "How to Set Conditionals" on page 772.
  - **Add** button: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

GS1-128 Settings

GS1-128 is also known as "EAN 128", "EAN/UCC 128" and "UCC 128". This barcode type not only encodes data, but also provides a mechanism for defining the meaning (or format) of that data. It supports alphanumeric data and some predefined Function Codes. See the Wikipedia GS1-128 entry for more information.

Note

To create dynamic barcodes, Metadata options must first have been set (see "Meta Data options" on page 699).
Metadata fields are required to create the association between the dynamic data used in the print run and the barcode.

Use the following options to configure the output Barcode settings:

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.

This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.

- **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.

- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the Barcode added to the page.
  - **Output once per sheet**: This option relates to Imposition printing (see Imposition), also known as N-Up printing. Select this box to have the Barcode printed once per sheet rather than once per document page.

    > **Note**
    >
    > If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

  - **Left**: Enter the distance between the left margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
  - **Bottom**: Enter the distance between the bottom margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.

- **GS1-128 Properties** group:
  - **Height**: Enter the Barcode height in either metric (cm/mm), inch (in) or point (pt) values.
  - **Module Width**: Specifies the width of the narrow bars. Changing this value to a higher value will generally make the Barcode bigger.
  - **Check Digit marker**: This character is used as a placeholder for the check digit, which we be calculated at runtime. The character must be expressed in Hex.
  - **Group separator**: This character is used to define group separation points. The character must be expressed in Hex.
  - **Template**: Specify an optional Barcode "template".
    Examples:

- `n13` defines a numeric field with exactly 13 digits.
- `n13+cd` defines a numeric field with exactly 13 digits plus a check digit.
- `an1-9` defines an alpha-numeric field with 1 to 9 characters.

Elements can be combined using the '+' symbol.

- **Print human readable text:** Check to add a textual version of the Barcode data.
- **Placement**: Use the drop-down to select whether to place the human readable text above or below the Barcode.
- **Font name**: Use the drop-down to select the font with which to display the human readable text.

> **Note**
>
> Vertically-oriented fonts (fonts whose typeface name begin with the at (@) character) are not supported in Connect.

- **Font size**: Enter a font size for the human readable text.

- **Process Tilde**:
  Check this option to process tilde characters in the data as special characters.

  Note: with this option checked, any tilde that needs to be included in the output must be escaped by adding another tilde: ~~.

- **Text**: Enter the text used to generate the Barcode.
  - **Add** button: Click to display a list of variable data that could be used for generating the Barcode.
    This includes metadata fields which must previously have been added in the Metadata options, likely at the *Document Tags* level (see Metadata Options), as well as some information fields.
- **Condition**: Enter the condition which determines whether or not the Barcode will be added to the document at print time. For details on how to create a conditional, see "How to Set Conditionals" on page 772.
  - **Add** button: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

EAN-13 Settings

EAN-13 barcodes are composed entirely of numerical data. The first 12 digits representing country/economic area, manufacturer and product codes + 1 following checksum digit.

> **Note**
>
> To create dynamic barcodes, Metadata options must first have been set (see "Meta Data options" on page 699).
> Metadata fields are required to create the association between the dynamic data used in the print run and the barcode.

Use the following options to configure the output Barcode settings:

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.
  - **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.
- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the Barcode added to the page.
  - **Output once per sheet**: This option relates to Imposition printing (see Imposition), also known as N-Up printing. Select this box to have the Barcode printed once per sheet rather than once per document page.

    > **Note**
    >
    > If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

- **Left**: Enter the distance between the left margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
- **Bottom**: Enter the distance between the bottom margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.

- **EAN 13 Properties** group:
  - **Height**: Enter the Barcode height in either metric (cm/mm), inch (in) or point (pt) values.
  - **Module Width**: Specifies the width of the narrow bars. Changing this value to a higher value will generally make the Barcode bigger. The EAN-13 barcode employs a module width between 0.27mm and 0.66mm.
    - **Checksum**: Use the drop-down to select how to deal with the Barcode checksum:
    - **Ignore**: Ignore checksum calculations.
    - **Auto**: Add a checksum character to the Barcode if the initial value does not validate. This is the default value.
    - **Check**: Verify the Barcode has a valid checksum.
    - **Add**: Calculate and add a checksum character to Barcode, regardless of current value.
    - **Print human readable text:** Check to add a textual version of the Barcode data.
    - **Placement**: Use the drop-down to select whether to place the human readable text above or below the Barcode.
    - **Font name**: Use the drop-down to select the font with which to display the human readable text.

      > **Note**
      >
      > Vertically-oriented fonts (fonts whose typeface name begin with the at (@) character) are not supported in Connect.

    - **Font size**: Enter a font size for the human readable text.

- **Process Tilde**:
  This selection is not applicable to this barcode type.

- **Text**: Enter the text used to generate the Barcode.
  - **Add** button: Click to display a list of variable data that could be used for generating the Barcode.
    This includes metadata fields which must previously have been added in the Metadata options, likely at the *Document Tags* level (see Metadata Options), as well as some information fields.
- **Condition**: Enter the condition which determines whether or not the Barcode will be added to the document at print time. For details on how to create a conditional, see "How to Set Conditionals" on page 772.
  - **Add** button: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

EAN-8 Settings

An EAN-8 barcodes are composed entirely of numerical data. It is comprised of 7 data digits containing the country/economic area code and an item reference code, with 1 following checksum digit.

Use the following options to configure the output Barcode settings:

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.
  - **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.
- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the Barcode added to the page.
  - **Output once per sheet**: This option relates to Imposition printing (see Imposition), also known as N-Up printing. Select this box to have the Barcode printed once per sheet rather than once per document page.

> **Note**
>
> If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

- **Left**: Enter the distance between the left margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
- **Bottom**: Enter the distance between the bottom margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.

- **EAN-8 Properties**:
  - **Height**: Enter the Barcode height in either metric (cm/mm), inch (in) or point (pt) values.
  - **Module Width**: Specifies the width of the narrow bars. Changing this value to a higher value will generally make the Barcode bigger. The EAN-8 barcode employs a module width between 0.27mm and 0.66mm.
  - - **Checksum**: Use the drop-down to select how to deal with the Barcode checksum:
    - **Ignore**: Ignore checksum calculations.
    - **Auto**: Add a checksum character to the Barcode if the initial value does not validate. This is the default value.
    - **Check**: Verify the Barcode has a valid checksum.
    - **Add**: Calculate and add a checksum character to Barcode, regardless of current value.
    - **Print human readable text:** Check to add a textual version of the Barcode data.
    - **Placement**: Use the drop-down to select whether to place the human readable text above or below the Barcode.
    - **Font name**: Use the drop-down to select the font with which to display the human readable text.

> **Note**
>
> Vertically-oriented fonts (fonts whose typeface name begin with the at (@) character) are not supported in Connect.

  - **Font size**: Enter a font size for the human readable text.

- **Process Tilde**:
  This selection is not applicable to this barcode type.

- **Text**: Enter the text used to generate the Barcode.
  - **Add** button: Click to display a list of variable data that could be used for generating the Barcode.
    This includes metadata fields which must previously have been added in the Metadata options, likely at the *Document Tags* level (see Metadata Options), as well as some information fields.
- **Condition**: Enter the condition which determines whether or not the Barcode will be added to the document at print time. For details on how to create a conditional, see "How to Set Conditionals" on page 772.
  - **Add** button: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

Interleaved 2 of 5 Settings

Interleaved 2 of 5 barcodes are also known as "ITF" and "2/5 Interleaved". It is a numeric only barcode whose data must contain an even number of digits, as the barcode uses sequences of two digits interleaved with each other to create a single symbol. If the numeric data contains an odd number of digits, then a leading zero must be added to the beginning of the data.

Use the following options to configure the output Barcode settings:

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have

included, at time of printing.

- **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.

- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the Barcode added to the page.
  - **Output once per sheet**: This option relates to Imposition printing (see Imposition), also known as N-Up printing. Select this box to have the Barcode printed once per sheet rather than once per document page.

    > **Note**
    >
    > If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

  - **Left**: Enter the distance between the left margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
  - **Bottom**: Enter the distance between the bottom margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.

- **Interleaved 2 of 5 Properties** group:
  - **Height**: Enter the Barcode height in either metric (cm/mm), inch (in) or point (pt) values.
  - **Module Width**: Specifies the width of the narrow bars. Changing this value to a higher value will generally make the Barcode bigger.
  - **Bar width ratio**: Set the Barcode bar width.
    - **Print human readable text:** Check to add a textual version of the Barcode data.
    - **Placement**: Use the drop-down to select whether to place the human readable text above or below the Barcode.
    - **Font name**: Use the drop-down to select the font with which to display the human readable text.

> **Note**
>
> Vertically-oriented fonts (fonts whose typeface name begin with the at (@) character) are not supported in Connect.

  - **Font size**: Enter a font size for the human readable text.

- **Process Tilde**:
  This selection is not applicable to this barcode type.

- **Text**: Enter the text used to generate the Barcode.
  - **Add** button: Click to display a list of variable data that could be used for generating the Barcode.
    This includes metadata fields which must previously have been added in the Metadata options, likely at the *Document Tags* level (see Metadata Options), as well as some information fields.
- **Condition**: Enter the condition which determines whether or not the Barcode will be added to the document at print time. For details on how to create a conditional, see "How to Set Conditionals" on page 772.
  - **Add** button: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

PDF417 Settings

PDF417 is a two-dimensional, multi-row Barcode. It is used for encoding large amounts of data, with hundreds or even thousands of characters. It encodes alphabetic text, numbers, binary files and actual data bytes.

> **Note**
>
> To create dynamic barcodes, Metadata options must first have been set (see "Meta Data options" on page 699).
> Metadata fields are required to create the association between the dynamic data used in the print run and the barcode.

Use the following options to configure the output Barcode settings:

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.
  - **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.
- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the Barcode added to the page.
  - **Output once per sheet**: This option relates to Imposition printing (see Imposition), also known as N-Up printing. Select this box to have the Barcode printed once per sheet rather than once per document page.

    > **Note**
    >
    > If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

  - **Left**: Enter the distance between the left margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
  - **Bottom**: Enter the distance between the bottom margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
- **PDF417 Properties** group:
  - **Module Width**: Specifies the width of the narrow bars. Changing this value to a higher value will generally make the Barcode bigger.
  - **Row height**: Defines the height of the bars for a single row, measured in pixels, points or metric.
  - **Width to height ratio**: Select the ratio of column width to row height.
  - **Mode**: Use the drop-down to set the compaction mode.

- **Binary**: allows any byte value to be encoded.
- **Text**: allows all printable ASCII characters to be end coded (ASCII values 32 to 126 and some additional control characters).
- **Numeric**: more efficient mode for encoding numeric data.
- **Auto**: Automatically detect the data content and encodes using the most appropriate method. This is the default option.

- **Error Correction Level**: Enter the error correction level for the built-in error correction method based on Reed-Solomon algorithms. The error correction level is adjustable between level 0 (just error detection, without correction) and level 8 (maximum error correction). Recommended error correction levels are between level 2 and 5, but the optimal value depends on the amount of data, printing quality of the PDF417 symbol and decoding capabilities.
- **Rows**: A PDF417 bar code can have anywhere from 3 to 90 rows.
- **Columns**: The number of data columns can vary from 1 to 30.

- **Tilde processing**

  Check this option to process tilde (~) characters in the data as special characters. (See the Java4less Barcodes Guide to learn what the tilde character can be used for.)
  The tilde is expected to be followed by the 'd' character and 3 digits representing an ASCII character: **~dNNN** . For example, ~d013 represents a carriage return.
  Note that with this option checked, any tilde that needs to be included in the output must be escaped by adding another tilde: ~~.

- **Text**: Enter the text used to generate the Barcode.
  - **Add** button: Click to display a list of variable data that could be used for generating the Barcode.
    This includes metadata fields which must previously have been added in the Metadata options, likely at the *Document Tags* level (see Metadata Options), as well as some information fields.
- **Condition**: Enter the condition which determines whether or not the Barcode will be added to the document at print time. For details on how to create a conditional, see "How to Set Conditionals" on page 772.
  - **Add** button: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

QR Code Settings

QR Code (Quick Response Code) is a 2D Barcode format that supports alphanumeric, numeric, byte/binary, and Kanji (Japanese-Chinese character) data.

> **Note**
>
> To create dynamic barcodes, Metadata options must first have been set (see "Meta Data options" on page 699).
> Metadata fields are required to create the association between the dynamic data used in the print run and the barcode.

Use the following options to configure the output Barcode settings:

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.
  - **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.
- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the Barcode added to the page.
  - **Output once per sheet**: This option relates to Imposition printing (see Imposition), also known as N-Up printing. Select this box to have the Barcode printed once per sheet rather than once per document page.

    > **Note**
    >
    > If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

- **Left**: Enter the distance between the left margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
- **Bottom**: Enter the distance between the bottom margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.

QRCode Properties group:

- **Size by**: Select size from the two options available:
  - **By area**: Connect will try to size the Barcode to fit the specified area by dynamically changing the module width to the **Size** selection. The lower module width limit is governed by the **Minimum module width** selection.
    Enter the sizes in either metric (cm/mm), inch (in), pixel (px) or point (pt) values.
  - **By module width**: Connect will try to size the Barcode to the module width of the characters. Large Barcode values will result in larger Barcode and vice versa.
    Enter the **Module width**in either metric (cm/mm), inch (in), pixel (px) or point (pt) values.
- **Encoding**: Define the encoding of the Barcode:
  - **Auto**: Automatically detect the data content and encodes using the most appropriate method. This is the default option.
  - **Numeric**: 7089 numerical characters.
  - **Alphanumeric**: 4296 alphanumerical characters.
  - **Byte**: 2953 characters.
  - **Kanji**: 1817 Japanese/Chinese characters.
- **Version**: Select the preferred QR code version (which sets the data length field) from the 40 available.

> **Note**
>
> The Encoding and Version fields work together to determine how many characters are encoded within a *length field*. The following table shows the number of bits in a length field, based upon the selections made:

Encoding	Ver. 1-9	Ver. 10-23	Ver. 27-40
Numeric	10	12	14

Alphanumeric	9	11	13
Byte	8	16	16
Kanji	8	10	12

- **Error Correction Level**: Part of the robustness of QR codes is their ability to sustain “damage” and continue to function even when a part of the QR code image is obscured, defaced or removed. A higher correction level duplicates data within the QR Code to allow for damaged areas. The higher the Error Correction Level, the larger the Barcode will be. The choices are (in order from lowest to highest): **L**ow, **M**edium, **Q**uartile and **H**igh.
- **Use ECI for encoding messages as bytes**: Selecting Extended Channel Interpretations (ECI) allows encoding multiple character sets (e.g. Arabic, Cyrillic, Greek, Hebrew) and other data interpretations, into one QR Code symbol.
- **Multi-part QR Code (structured append)**: Select to append a QR Code symbol in a structured format.
  - **Part**: indicates the position of the QR Code symbol within the group of Structured Append symbols.
  - **of**: indicates how many Structured Append symbols exist.

> **Note**
>
> The Structured Append symbols Part number can never exceed the sum total of Structured Append symbols available (the "of" value). Thus selecting a Part number beyond the existing sum total will increase the sum total to the same value.

- **Use FNC1**: Check to enable Application Identifiers. These are often used to encode links to websites, or to encode production/batch details.
  - **Position**: Select between the two methods for encoding FNC1 characters within QR Codes:
    - **First** Position - uses the GS1 QR Code standard.
    - **Second** Position - uses the AIM QR Code standard. If this option is chosen then the appropriate Application Indicator will also need to be set.

- **Application ID**: Enter the appropriate QR-Code Application Indicator in accordance with the specific industry or application specifications (as provided by AIM International).

- **Process Tilde**:
  Check this option to process tilde characters in the data as special characters.

  Note: with this option checked, any tilde that needs to be included in the output must be escaped by adding another tilde: ~~.

- **Text**: Enter the text used to generate the Barcode.
  - **Add** button: Click to display a list of variable data that could be used for generating the Barcode.
    This includes metadata fields which must previously have been added in the Metadata options, likely at the *Document Tags* level (see Metadata Options), as well as some information fields.
- **Condition**: Enter the condition which determines whether or not the Barcode will be added to the document at print time. For details on how to create a conditional, see "How to Set Conditionals" on page 772.
  - **Add** button: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

UPC-A Settings

The Universal Product Code (UPC-A) Barcode is widely used for tracking trade items in stores and at the point-of-sale. It consists of 12 numerical digits which are uniquely assigned to each trade item.

> **Note**
>
> To create dynamic barcodes, Metadata options must first have been set (see "Meta Data options" on page 699).
> Metadata fields are required to create the association between the dynamic data used in the print run and the barcode.

Use the following options to configure the output Barcode settings:

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.
  - **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.
- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the Barcode added to the page.
  - **Output once per sheet**: This option relates to Imposition printing (see Imposition), also known as N-Up printing. Select this box to have the Barcode printed once per sheet rather than once per document page.

    > **Note**
    >
    > If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

  - **Left**: Enter the distance between the left margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
  - **Bottom**: Enter the distance between the bottom margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.

UPC A Properties group:

- **Height**: Enter the Barcode height in either metric (cm/mm), inch (in) or point (pt) values.
- **Module Width**: Specifies the width of the narrow bars. Changing this value to a higher value will generally make the Barcode bigger.
  - **Checksum**: Use the drop-down to select how to deal with the Barcode checksum:

- **Ignore**: Ignore checksum calculations.
- **Auto**: Add a checksum character to the Barcode if the initial value does not validate. This is the default value.
- **Check**: Verify the Barcode has a valid checksum.
- **Add**: Calculate and add a checksum character to Barcode, regardless of current value.
- **Print human readable text:** Check to add a textual version of the Barcode data.
- **Placement**: Use the drop-down to select whether to place the human readable text above or below the Barcode.
- **Font name**: Use the drop-down to select the font with which to display the human readable text.

> **Note**
>
> Vertically-oriented fonts (fonts whose typeface name begin with the at (@) character) are not supported in Connect.

- **Font size**: Enter a font size for the human readable text.

- **Process Tilde**:
  This selection is not applicable to this barcode type.

- **Text**: Enter the text used to generate the Barcode.
  - **Add** button: Click to display a list of variable data that could be used for generating the Barcode.
    This includes metadata fields which must previously have been added in the Metadata options, likely at the *Document Tags* level (see Metadata Options), as well as some information fields.
- **Condition**: Enter the condition which determines whether or not the Barcode will be added to the document at print time. For details on how to create a conditional, see "How to Set Conditionals" on page 772.
  - **Add** button: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

UPC-E Settings

The Universal Product Code (UPC-E) Barcode is widely used for tracking trade items in stores and at the point-of-sale. It consists of 6 numerical digits which are uniquely assigned to each trade item.

> **Note**
>
> To create dynamic barcodes, Metadata options must first have been set (see "Meta Data options" on page 699).
> Metadata fields are required to create the association between the dynamic data used in the print run and the barcode.

Use the following options to configure the output Barcode settings:

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.
  - **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.
- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the Barcode added to the page.
  - **Output once per sheet**: This option relates to Imposition printing (see Imposition), also known as N-Up printing. Select this box to have the Barcode printed once per sheet rather than once per document page.

    > **Note**
    >
    > If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

- **Left**: Enter the distance between the left margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.
- **Bottom**: Enter the distance between the bottom margin of the page and the Barcode, in either metric (cm/mm), inch (in) or point (pt) values.

UPC A Properties group:

- **Height**: Enter the Barcode height in either metric (cm/mm), inch (in) or point (pt) values.
- **Module Width**: Specifies the width of the narrow bars. Changing this value to a higher value will generally make the Barcode bigger.
  - **Checksum**: Use the drop-down to select how to deal with the Barcode checksum:
  - **Ignore**: Ignore checksum calculations.
  - **Auto**: Add a checksum character to the Barcode if the initial value does not validate. This is the default value.
  - **Check**: Verify the Barcode has a valid checksum.
  - **Add**: Calculate and add a checksum character to Barcode, regardless of current value.
  - **Print human readable text:** Check to add a textual version of the Barcode data.
  - **Placement**: Use the drop-down to select whether to place the human readable text above or below the Barcode.
  - **Font name**: Use the drop-down to select the font with which to display the human readable text.

    > **Note**
    >
    > Vertically-oriented fonts (fonts whose typeface name begin with the at (@) character) are not supported in Connect.

  - **Font size**: Enter a font size for the human readable text.
- **Process Tilde**:
  This selection is not applicable to this barcode type.

- **Text**: Enter the text used to generate the Barcode.
  - **Add** button: Click to display a list of variable data that could be used for generating the Barcode.
    This includes metadata fields which must previously have been added in the Metadata options, likely at the *Document Tags* level (see Metadata Options), as well as some information fields.
- **Condition**: Enter the condition which determines whether or not the Barcode will be added to the document at print time. For details on how to create a conditional, see "How to Set Conditionals" on page 772.
  - **Add** button: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

Additional OMR Mark Settings

The **Add OMR** dialog displays the properties of an OMR Mark that was added in the "Additional Content" on page 732 page.

These OMR marks differ from High Capacity Feeder (HCF) generated inserter marks. Those marks are specific to the inserter machine they were created for, whereas these additional OMR marks are completely independent and customizable. These custom OMR marks can be used to cater for inserter machines not currently supported by a HCF, or they can be used for any non-inserter related post-processing driven by OMR marks.

- **General** group:
  - **Include in output** checkbox: This determines whether of not this Additional Content entry should be included in the output or not.
    It is checked by default for all new entries.
    This option allow for "libraries" of Additional Content to be created in Presets, libraries from which you can pick and choose what entries you wish to have included, at time of printing.
  - **Description** edit box: An edit box for adding an optional Description to the Additional Content entry.
- **Position** group:
  - **Orientation**: Use the drop-down to select the orientation of the OMR Mark added to the page.
  - **Page Side**: Select whether the OMR Mark will print on the front or back of page.

- **Output once per sheet**: Option relates to Imposition (also known as N-Up) printing. Select this box to have the OMR Mark printed once per sheet rather than once per document page.

  > **Note**
  >
  > If Imposition options such as auto-positioning and scaling were selected, these options won't apply to the Additional Content added to the physical N-Up sheet.

- **Left**: Enter the distance between the left margin of the page and the OMR Mark, in either metric (cm/mm), inch (in), pixel (px) or point (pt) values.
- **Bottom**: Enter the distance between the bottom margin of the page and the OMR Mark, in either metric (cm/mm), inch (in), pixel (px) or point (pt) values.

- **Options Tab**:
  - **Collation Level**: Choices are:
    - **Document**: Treats each document as a group and the group and match marks will be set based upon the start and end of a document.
    - **Document Set**: Treats each document set as a group and the group and match marks will be set based upon the start and end of a document set.
  - **Draw Hot Spots**: This adds a red rectangle around the location of each individual mark in the output, allowing easier checking of the OMR mark logic.
  - **Line Options** group:
    - **Line Thickness**: Sets the thickness of each OMR mark line.
    - **Line Length**: Sets the lenght of each OMR mark line.
    - **Line Spacing**: Determines how the spacing between each OMR mark line will be set. The associated control beneath the combination box will be enabled, based upon this selection.
      - **Line Per Inch**: If **Line Spacing** is set to *Lines Per Inch* this option will be enabled. It defines how many lines will print per inch.
      - **Gap Distance**: If **Line Spacing** is set to *Gap Distance* this option will be enabled. It defines the size of the gap between lines; i.e. the distance from the bottom of one OMR mark line to the top of the next.

- **Line Distance**: If **Line Spacing** is set to *Line Distance* this option will be enabled. It defines the distance from the top of one line to the top of the next.

- **Sequence Number Range** group: Allows selection of Start and Stop points for the wrapping page sequence number in a group.
  For example, a range of 2-10 would cause the sequence numbers to iterate as follows: 2, 3, 4, 5, 6, 7, 8, 9, 10, 2, 3, 4 ...

  > **Note**
  >
  > The Sequence Number iterates per page within a group and is used to identify missing pages in a group.

  - **Start**: The starting point for the range
  - **Stop**: The end point of the range
  - **Start number**: The number to start from (from within the selected range).
- **Match Number Range** group: Allows selection of Start and Stop points for the wrapping match number for a group.
  For example, a range of 1-6, with a Start number of 2 would cause the matched numbers to be as follows: 2, 3, 4, 5, 6, 2, 3, 4, 5, 6, 2, ...

  > **Note**
  >
  > The Match Number iterates per group and is used to identify missing groups

  - **Start**: Start number
  - **Stop**: Stop number
- **Condition**: Enter the condition which determines whether or not the OMR Mark will be added to the document at print time.
  For details on how to create a conditional, see the "How to Set Conditionals" on page 772 page.

- **OMR Marks Tab**:

- **#**: OMR Mark number (display only).
- **Type**: Type of OMR Mark (display only).
- **Value**: OMR Mark Value. These can be selected and altered for Sequence, Match and Parity marks, as described below.
- **Add**: Add an OMR Mark entry to the table.
  Choices are between:
  - **On**: This represents a mark that is always printed.
  - **Off**: This represents a mark that is never printed; i.e. it pads the marks out with an empty position.
  - **Group Start**: This represents a mark that is printed on the first page of a group.
  - **Group End**: This represents a mark that is printed on the last page of a group.

    Note

    In a single page group both Group Start and End marks will print if defined since the page is both the start and end of the group.

  - **Sequence**: This represents a mark that is printed when the specified bit is set in the sequence number of the page.
    For example, if the bit for the mark is set to 2 and the sequence number for the page is 5 then it will not print since the value 5 consists of the bits 1 and 4.
    Use the drop down box to select the entry.
  - **Match**: This represents a mark that is printed when the specified bit is set in the match number of the group.
    For example, if the bit for the mark is set to 2 and the match number for the group is 3 then it will print since 3 consists of the bits 1 and 2.
    Use the drop down box to select the entry.

    Note

    The match number is the same for all pages in a group.

- **Parity**: This mark prints in order to maintain the parity of the number of lines printed on the page. If set to *Even* then it will print if the total count of the other printed marks in the printed is odd.
  For example, by printing the parity mark it will create an even number of marks on the page. And vice versa with *Odd* parity - the parity mark will print if the total number of other printed marks on the page is even in order to keep the overall count odd.
  Use the drop down box to select the entry.
- **Conditional**: Enter the condition which determines whether or not this OMR Mark will be added to the document at print time.
  For details on how to create a conditional, see the "How to Set Conditionals" below page

- **Delete**: Delete an entry from the table
- **Move up**: Move a entry up the table
- **Move down**: Move a entry down the table
- **Edit**: Edit a **Conditional** entry within the table.

> **Tip**
>
> You can also double click a **Conditional** entry within the table to edit it.

How to Set Conditionals

- **Condition**: Enter the condition which determines whether or not this element will be added to the document at print time.

  Use JavaScript conditional expressions to construct your conditional.
  You may use JavaScript logical operators (==, !=, <, >, &&, ||, etc), literal values ("text', 3.14, etc), mathematical operators (/, *, +, -, etc) and data fields.

  - **Add**: Click to add a field to the to the conditional expression.
    Select from a list of metadata fields (as added in Metadata Options), document information fields (see Variables Available in Output), or common expressions.

    For example, selecting the **Expressions > First Page** would add "page.nr == 1" to the conditional entry.

Example: if (page.nr == 1) { ... }

Booklet Options

This page appears as part of the **Advanced Print Wizard** and the Output Creation Preset Wizard.

The **Booklet Options** page defines how to generate booklets in the output. It is used in conjunction with Imposition settings, which will appear after the Booklet entries have been made.

This page includes a handy illustration that displays how the final binding would look, based upon the current selections.

Options:

- **Configuration**: Use the drop-down to select the type of binding to use:
  - **Saddle Binding**: This binding places all the pages in a stack, binds the middle and folds the stack as one.
  - **Perfect Binding**: This binding type is often used for books. Pages are folded in the middle and then set side by side. The pages are then bound along the folded "spine".
  - **1 up Perfect Binding**: This binding does not contain any folding. The pages are lined up side by side and bound along one edge.
- **Booklet Binding Edge**: Use the drop-down to select the side on which to bind the booklet.

Optional **Cover Page** selections are available to Saddle Binding only.

- **Cover Page** checkbox: Check to enable cover pages to be created with the options below:
  - **Media** selections:
    - **Cover Media Size**: Use the drop-down to select the media size for the cover page, or use a Custom size and select **Width** and **Height** values.
  - **Front Cover** selections:

- **Blank**: Select to add no data to the front cover.
- **First page on outside and second page on inside**: Select to use the first 2 pages as the inside and outside of the front cover.

- **Back Cover** selections:
  - **Blank**: Select to add no data to the back cover.
  - **Last two pages on inside and outside**: Select to use the final 2 pages as the inside and outside of the back cover.

Advanced Print Wizard and Output Creation Preset Wizard navigation options

- **Preview** button (active in ***Advanced Print*** Wizard only): Click to launch a Proof Preview window, which displays how the printed output would look based upon the currently chosen selections.
- **Back** and **Next** buttons: Used to navigate back and forth through all the selected options within the Wizard. Up until the Print button is pressed, one can reverse all the way through the wizard to return to the main selection page (the "Print options" on page 704 page) and add or remove printing options from the print run.
- **Print** button (***Advanced Print*** Wizard only) or **Finish** button (***Output Creation Preset*** Wizard only):
  Click to produce print output/finalize the Preset according to the current settings.
  This can be done at any point within the Wizard, whether or not *all* the options selected in the "Print options" on page 704 page have been completed or not.
- **Cancel** button: Cancels the Print Wizard, without creating any printout.

Imposition options

This page appears as part of the **Advanced Print Wizard** and the Output Creation Preset Wizard.

Imposition refers to the printing of multiple pages on a single sheet. This is also known as N-Up printing.

The options on the **Imposition Options** page allow for the setting of imposition repetition, order, margins and markings.

As imposition selections are very specific and can be quite confusing, we have provided a handy dynamic diagram that displays a representation of the current imposition selections in real time. Whenever selections are changed, this display changes to reflect the selections

made.

You can even select how many pages are to be represented, whilst page numbers appear on the pages in the diagram and change dynamically, like everything else.

It is important to note that if Impositioning is used then the only meta data that will be available to the job thereafter will be the meta data at Job Segment level.

When using Impositioning everything within the Segment is Impositioned, so all meta data below the Job Segment level is lost. If *Booklet Binding* were selected, some of the imposition settings will be determined by the options made within the "Booklet Options" on page 773 Page and those imposition settings will thus be disabled on this page.

Note

During Impositioning, all hyperlinks - both external and internal - are removed from the document (see "Hyperlink and mailto link" on page 268).

The Imposition selections that can be made are as follows:

- **Sample Page** group: Use to choose a paper size for the logical page(s) which would be printed upon the actual physical Imposition sheet. This helps in decision making regarding what paper sizes and orientations to use, with the Sample Page shown in blue on the white Imposition Sheet in the output preview (and validation) diagram.

  Note

  The logical page size can be set independently of any template open in the Designer, allowing for output presets to be created independently from the active template.

  If a template is currently loaded in Connect Designer, the size and portrait attribute of the first section of that template are used as defaults for the Sample Page.

  As a general rule, it makes little sense to modify the Sample Page size within the

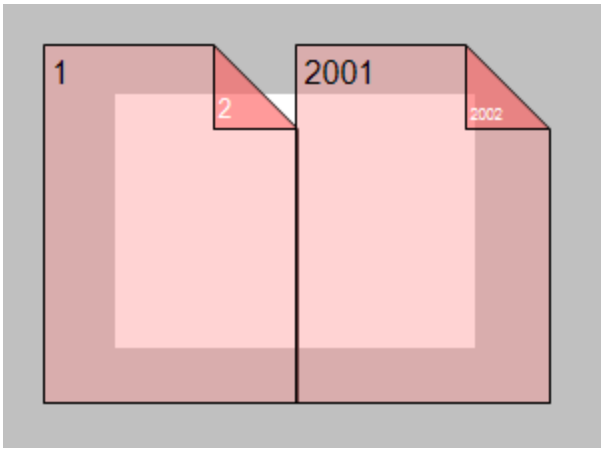
Print Wizard (as the template media size will already have been selected), but it does make sense when creating Print Output Presets for other templates.

If no template is loaded or if the template does not have a Print section, then the Sample Page defaults to A4 portrait.

  - **Size**: Use the drop-down to select the media size of the logical template page.
  - **Orientation**: Select orientation (aspect ratio) of the logical template page (*Landscape* or *Portrait*).
  - The **Width** and **Height** values are real-only and appear for display and design purposes only.
- **Imposition Sheet** group: Use to select the physical media size that the logical pages will be printed upon. This appears as the white background in the output preview (and validation) diagram.
  - **Size**: Use the drop-down to select the size of the physical media the output is to be printed upon.
  - **Orientation**: Select orientation (aspect ratio) of physical media sheet (*Landscape* or *Portrait*).
  - The **Width** and **Height** values are real-only and appear for display and design purposes only.
- Output diagram and validation. Use the settings here to control how the output diagram is to look and behave. It contains the following options:
  - **Display size** entry: Chose between a pre-set viewing percentage, or manually enter whatever percentage you would like, or select *Fit Page* or *Fit Width* to have the diagram display the page(s) within the diagram window dimensions.
  - Use the zoom in/out buttons / to increase or reduce the viewing percentage by 25% increments.
  - **Sample size** entry: Set the sample output size here. The output diagram will change dynamically to reflect the selection made here, if applicable.
    The maximum Sample size is 10,000 for booklet imposition and 10,000,000 for standard impositions.
  - Output diagram: A real-time display that reflects the selections made in the Imposition Options page. The blue page(s) are the individual logical page(s), upon

a white background that represents the physical Imposition Sheet.

If the settings are invalid, the display will appear as red, such as in the following diagram:



In this example the error is that the logical pages are too large to fit upon the Imposition Sheet that has been selected.

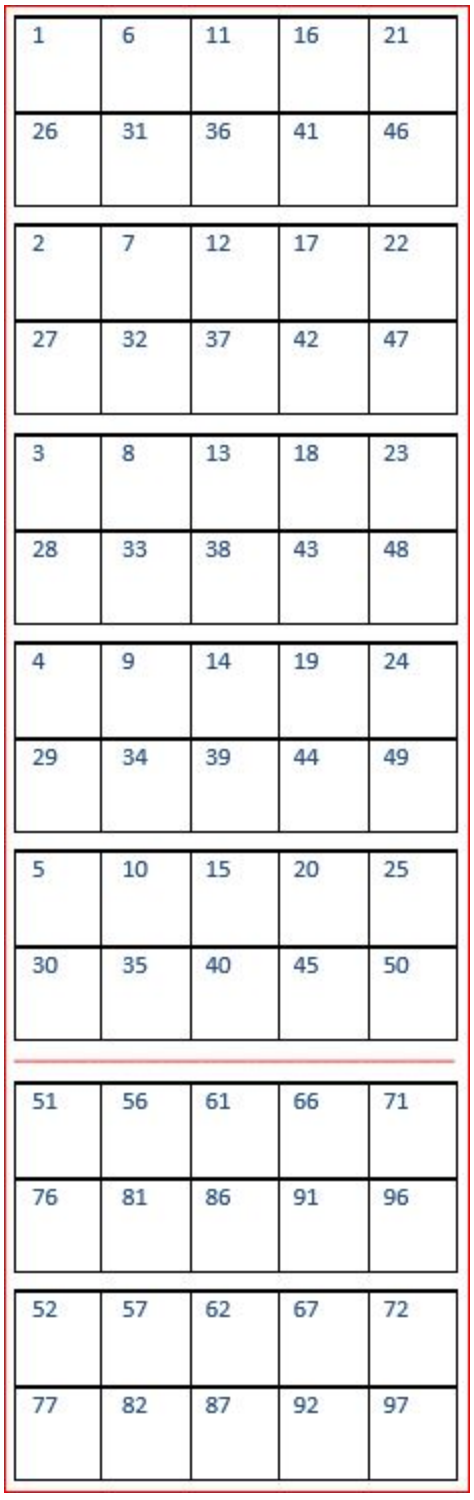
- **Position** group: These settings apply to the positioning of the logical page(s) within the Imposition Sheet.
  - **Method** selection: Set the position of the logical page(s) upon the Imposition Sheet. The choices are as follows:
    - *Center*: Have the logical page(s) centered within the Imposition Sheet.
    - *Scale to fit*: Scales the logical page(s) so that they fit the Imposition Sheet exactly.
    - *Offset*: Have the logical page(s) offset within the Imposition Sheet. The offset values can be set in the Left and Top entry boxes, which only become available for this positional method.
  - **Left** and **Top** values: These entry boxes are only enabled and their values applied when the Offset method has been chosen. They are used to set the offset of the logical page(s) upon the Imposition sheet.
  - **Rotate** selection: Select to rotate the logical page(s) within the Imposition Sheet.
- **Layout** group: Select how the logical page(s) are to be laid out on the Imposition Sheet.
  - **Horizontal** selection: Set how many logical pages are to be placed on one horizontal line within the Imposition Sheet.

> **Note**
>
> If the logic page was set to **Rotate** in the Position selections, then the Horizontal selection might appear as vertical, and the Vertical selection might appear as horizontal, on the real time output diagram.

- **Vertical** selection: Set how many logical pages are to be placed on one vertical line within the Imposition Sheet.
- **Horizontal Gap** selection: Set the amount of blank space to add between each logical page on this axis.
- **Vertical Gap** selection: Set the amount of blank space to add between each logical page on this axis.
- **Stack Depth** selection: Stack Depth defines the number of Imposition Sheets that will be printed before drill sorting resets.

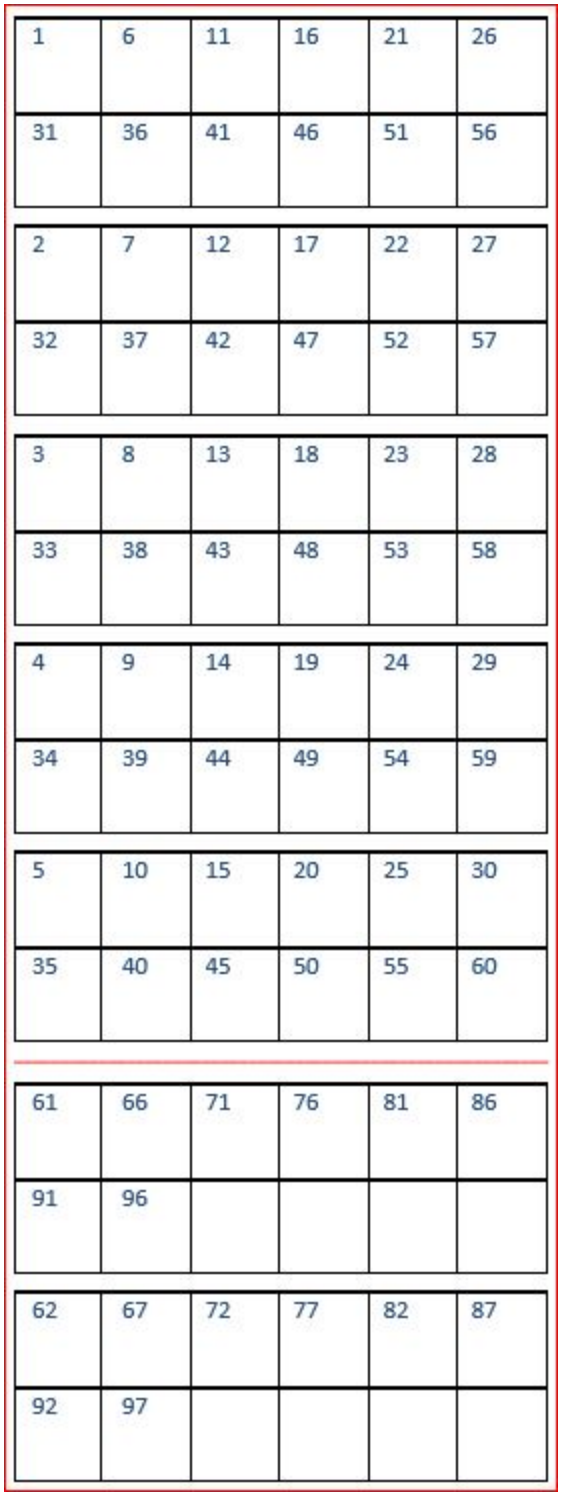
  Stack Depth works in conjunction with the Horizontal and Vertical page, Gap and Page Order selections in order to determine the printing order. This is best illustrated by way of example.

  The following "Cut and stack" example has a job with 100 single page documents and repetition settings of 5 Horizontal and 2 Vertical, plus a Stack Depth of 5. Each table represents a single Imposition Sheet, and the red line between the tables represents the Stack Depth:



Each set of 5 Imposition Sheets is a discrete drill sorted nUp block that can be guillotined and stacked. The first 5 sheets encompass records 1-50, whilst the next 5 sheets encompass records 51-100.

The following example is the same job but with a Horizontal repetition setting of 6:



1	6	11	16	21	26
31	36	41	46	51	56
2	7	12	17	22	27
32	37	42	47	52	57
3	8	13	18	23	28
33	38	43	48	53	58
4	9	14	19	24	29
34	39	44	49	54	59
5	10	15	20	25	30
35	40	45	50	55	60
61	66	71	76	81	86
91	96				
62	67	72	77	82	87
92	97				

As you can see, the outputs are *very* different.

Note

If the Stack Depth is greater than the number of sheets in the entire job then you will get a single drill sorted block that can be guillotined and stacked to give the complete job in record order.

- **Sides** selection: Select how many sides of the Imposition Sheet are to be printed upon. Choice is between *Simplex* or *Duplex*.
- **Method** selection: In 2018.2 we introduced a new imposition page order for cut & stack impositioning that works with continuous feed devices: *Stack by Column*. This option is intended for impositions with more than 1 row. It places the next page in the next row of the same column on the current sheet, instead of in the same row and column of the next sheet like the pre-existing *Cut and Stack* does.
  The choices are between:
  - *Cut and Stack*: Any preset created in a version of Connect prior to Connect 2018.2 will default to this Layout Method. It allows total control of subsequent **Page Order** choices.
  - *Stack by Column*: For use with continuous feed printers. This selection limits subsequent **Page Order** choices to column based options only.
- **Page Order**: Select in which direction to go when adding sections to the output:
  - **Left to right, top to bottom**
  - **Right to left, top to bottom**
  - **Top to bottom, left to right**
  - **Top to bottom, right to left**
- **Reverse** selection: The reverse selections have been extended in Connect 2018.2, with the addition of the *Inverse page order* option. The options now available are:

- *Off*. Output is not reversed.
- *Stack upside down*: Essentially turns the stack upside down by reversing the output imposition page sides. The last sheet becomes the first, and in the case of duplex imposition, the back sides become the front sides.
  For presets created in a version prior to Connect 2018.2, a "Reverse Pages" selection will apply this *Stack upside down* setting.
- *Inverse page order*: This impositions the output in the ***exact opposite order*** from what has been selected.
  For example, if the selected order was **Top to bottom, left to right** (down the stack, left to right, top to bottom), then selecting **Inverse page order** would result in output that was up the stack, right to left, bottom to top.
  In the case of duplex impositioning, the front and back sides don't switch.

- **CropMarks** group: Select to add Crop Marks to the printed output.
  - **Type**: Use the drop-down to select whether to add Crop Marks to the page or not, and what type of Crop Marks to use. The Crop Mark types available are *Standard* or *Japanese*.
  - **Page side**: What side(s) of the page to put the Crop Marks. The choices are *Front*, *Back* or *Both*.
  - **Width**: Select the width of the crop mark lines.
  - **Offset**: How much separation (if any) to leave between the vertical and horizontal corner markings.
  - **Length**: Select the Length of the crop mark lines.

Advanced Print Wizard and Output Creation Preset Wizard navigation options

- **Preview** button (active in ***Advanced Print*** Wizard only): Click to launch a Proof Preview window, which displays how the printed output would look based upon the currently chosen selections.
- **Back** and **Next** buttons: Used to navigate back and forth through all the selected options within the Wizard. Up until the Print button is pressed, one can reverse all the way through the wizard to return to the main selection page (the "Print options" on page 704 page) and add or remove printing options from the print run.
- **Print** button (***Advanced Print*** Wizard only) or **Finish** button (***Output Creation Preset*** Wizard only):
  Click to produce print output/finalize the Preset according to the current settings.

This can be done at any point within the Wizard, whether or not *all* the options selected in the "Print options" on page 704 page have been completed or not.

- **Cancel** button: Cancels the Print Wizard, without creating any printout.

Inserter options

This page appears as part of the **Advanced Print Wizard** and the Output Creation Preset Wizard.

The **Inserter Options** page allows the selection of a High Capacity Feeder (HCF) model. These machines are also commonly referred to as Inserters or Folder-Inserters.

The options available on this page are dependent upon the model selected.

The options selected on this screen influence the position of the markings set on the next page: **"Mark Position Options" on page 785**.

- **Model**: Use the drop-down to select from any previously loaded Inserter model, or use the Browse button to select a HCF file to load a new Inserter model.
  An image representing the chosen folder-inserter is displayed under the list, along with the HCF file details.
- **Options Group:**
  The options available here are all Inserter dependent, and thus will change based upon the Inserter model selection.
  To see how the selected Inserter markings would look on the printed page, click the Next button to move to the "Mark Position Options" on page 785 page, which has a preview of the page. You can move back and forward between these two pages until you are entirely satisfied with the selections made.
  - **Mark Configuration**: Use the drop-down to select the type of markings to add. This selection basically equates to the amount of area the markings will take up on the printed page.
  - **Fold Type**: Use the drop-down to select the type of fold to apply to the paper. This will impact upon where on the page the markings will be placed.
  - **Collation level**: Select whether the markings will be made at Document level, or Document Set level.
  - **Print marks on back**: Check to place the Inserter Marks on the rear of the page.
  - **Selective Inserts**: If selective inserts are supported by the chosen Mark Configuration you can select what markings to include and whether those markings

are to included based upon some conditional setting. This can be done by highlighting the Mark Name entry and either pressing the **Edit** button, or using the right mouse click context menu, and selecting **Edit**.
For information on how to edit the Selective Inserts settings, see "Selective Insert Dialog" on page 786

- **Clear Background Area** Tab:
  Check the **Clear Background Area** checkbox to add a white background to the OMR, preventing background colors or elements interfering with the OMR Markings when they are read by the Inserter.
  - **Margins**:
    - **Same for all sides**: Check so that the Left margin selection is used to set all sides identically.
    - **Left, top, right, bottom**: Enter measurements for the margins on each side of the OMR Marks.
- **Custom OMR mark sizing** Tab:
  If supported by the currently chosen Mark Configuration you can select a Custom OMR size by checking the **Custom OMR mark sizing** checkbox.
  Select from any of the following, or leave the entries blank to use default values:
  - **Line length**: Enter a value between 10.16mm and 20mm.
  - **Line thickness**: Enter a value between 0.254mm and 0.63mm.
  - **Gap distance**: Enter a millimeter value 2.91mm and 4.2mm.

Advanced Print Wizard and Output Creation Preset Wizard navigation options

- **Preview** button (active in ***Advanced Print*** Wizard only): Click to launch a Proof Preview window, which displays how the printed output would look based upon the currently chosen selections.
- **Back** and **Next** buttons: Used to navigate back and forth through all the selected options within the Wizard. Up until the Print button is pressed, one can reverse all the way through the wizard to return to the main selection page (the "Print options" on page 704 page) and add or remove printing options from the print run.
- **Print** button (***Advanced Print*** Wizard only) or **Finish** button (***Output Creation Preset*** Wizard only):
  Click to produce print output/finalize the Preset according to the current settings.
  This can be done at any point within the Wizard, whether or not *all* the options selected in

the "Print options" on page 704 page have been completed or not.

- **Cancel** button: Cancels the Print Wizard, without creating any printout.

Mark Position Options

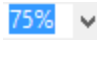
This page appears as part of the **Advanced Print Wizard** and the Output Creation Preset Wizard.

The **Mark Position Options** page displays a Preview of the output and the possible locations to place the inserter marks. The initial settings are determined by the selections made within the "Inserter options" on page 783 page.
You can move back and forward between these two pages to perfect the settings, or you could move the inserter mark box to the desired location on the preview.

Preview box:

- The *pink area* displays the areas of the page where inserter marks can be positioned.
- The *small checkered box* displays the current location of the inserter marks. This box is selectable and can be dragged to the desired location within the printable (pink) areas. If the box is placed outside the printable areas the page will display an error and prevent attempts at leaving the page.

Below the Preview box are buttons which allow control of the Preview box. The selections that can be made are:

- **First Page**: Click to jump to the first page.
- **Previous Page**: Click to move to the previous page.
- **Next Page**: Click to move to the next page.
- **Last Page**: Click to jump to the last page.
- **Show Page**: Use the up and down arrows or type a page number to display a specific page within the document.
- **Zoom in/out**: Click to zoom in or out by 25%
- 

Zoom Level: Use the drop-down to select a predefined level or enter a zooming percentage.

Selective Insert Dialog

The Selective Insert Dialog provides control over individual inserts. In this dialog the selected can be set to be either selected ("Yes") or not ("No"), or selected based upon some conditional criteria ("Conditional").

For example, you could add a marking to the third page of a document by making the selection Conditional and then setting the Condition entry to "page.nr == 3".

Both Metadata and Informational fields can be used in the Conditionals settings.

For more details on how to create a conditional, see the "How to Set Conditionals" on page 772 page.

Printer settings

This page appears as part of the **Advanced Print Wizard** and the Output Creation Preset Wizard.

The **Printer Settings** page provides options for cut-sheet printers. It maps media types to printer trays.
It is available for AFP, IPDS, PCL and PostScript printers that are configured for cut-sheet printing.

> **Note**
>
> Not all the settings will be available for all printers. Only those printers whose Printer Definition supports the extra information will have the **Position**, **Weight**, **Type** or **Color** options available.

- **Map media by** options: Select from following choices:
  This entry will only be available for those printers that support the extra information.
  - **Media Attribute** displays all Media details, except the Tray selection.
  - **Tray** displays just the Media name and Tray selections.
  - **Both** displays all Media details.
- **Media/Tray Table** columns:

- **Media**: Lists the Media name, as defined in the template.
- **Tray**: Use the drop-down to select in which Tray to send any page using the media.
- **Position**: Enter a MediaPosition option on the printer to define the media to use. This entry will only be available for those printers that support the extra information.
- **Weight**: Enter a weight for the paper. This entry will only be available for those printers that support the extra information.
- **Type**: Use the drop-down to select which type of stock to use on the printer. This entry will only be available for those printers that support the extra information.
- **Color**: Use the drop-down to select which color the paper should be on the printer. This entry will only be available for those printers that support the extra information.

- **Media/Tray Table** buttons:
  - **Add**: Adds a new Media/Tray entry in the table.
  - **Delete**: Deletes the current Media/Tray selections from the table.
  - / **Move Up / Move Down**: Move the selected Media/Tray entries up or down within the table.
  - **Import Tray Settings**: Import the Media/Tray settings from a Connect Template.

    This entry will only be available for those printers that support the extra information. The options are to:
    - **Import from current template**, which will import any Media/Tray settings from the current Connect Template into the table.
    - **Import from template file**, which allows you to browse for a Connect Template to import the Media/Trays settings from.

    There is no restriction on how many Templates you may import settings from.
- **Output Destination** group: These settings allow for the selection of output bins. It is available for AFP, IPDS and PCL printers whose Printer Definition caters for multiple output bins. If the selected Printer Definition does not support bins, this group will not be visible on the page.

  The table has the following choices:
  - **Trigger**: Select what the trigger will be for changing bins. The options are *>Page*, *>Sheet*, *>Document*, *>Document Set* or *>Job Segment*.
  - **Destination**: Select which bin should be used when this trigger is met. The options will be the bins available to the selected Printer Definition.

- **Rules**: Defaults to true, but can be altered at will. Double clicking in the field brings up the Rule Editor, which can be used to construct the rule(s) that determines whether this bin is to be used or not.

The options (both button and right mouse click context menu) available to the table are:

- **Add**: Adds a new bin selection choice to the table.
- **Delete**: Deletes the current selection(s) from the table.
- / **Move Up / Move Down**: Move the selected entries up or down within the table.
- **Edit ...**: Edit the current selection.

Default output destination selection: Select the default output bin from those bins available to this Printer Definition.

Advanced Print Wizard and Output Creation Preset Wizard navigation options

- **Preview** button (active in ***Advanced Print*** Wizard only): Click to launch a Proof Preview window, which displays how the printed output would look based upon the currently chosen selections.
- **Back** and **Next** buttons: Used to navigate back and forth through all the selected options within the Wizard. Up until the Print button is pressed, one can reverse all the way through the wizard to return to the main selection page (the "Print options" on page 704 page) and add or remove printing options from the print run.
- **Print** button (***Advanced Print*** Wizard only) or **Finish** button (***Output Creation Preset*** Wizard only):
  Click to produce print output/finalize the Preset according to the current settings.
  This can be done at any point within the Wizard, whether or not *all* the options selected in the "Print options" on page 704 page have been completed or not.
- **Cancel** button: Cancels the Print Wizard, without creating any printout.

Dynamic PPD Options

This page appears as part of the **Advanced Print Wizard** and the Output Creation Preset Wizard.

The **Dynamic PPD mode options** page allows you to add PostScript printer capabilities to Connect, through the printer's associated PostScript Printer Definition (PPD) file, and to assign rules to them to trigger those capabilities in conjunction with Connect Templates.

Note

This Print Wizard page is intended for use only by those operators who are very familiar with their PostScript printer's capabilities and the options available to them in the printer's associated PPD file.

A video example of how to set Dynamic PPD Options can be found here: Setting Dynamic PPD Options within Connect - An example.

Note

As of OL Connect 2019.1 *JobPatchFile* sections in PPDs are automatically added to the output, if the PPD includes them.

The **Dynamic PPD mode Options** page can be broken up as follows:

PPD Info group

This shows information about the selected PostScript Printer Definition. The information box contains general printer details (such as the model name) whilst the **PPD file name** field displays the name of the actual PPD file that is loaded.

If the incorrect PPD was assigned to this Printer Model, the PPD file can be changed at this point, via the **Import PPD** button .

This is not considered good practice, however. It would be better to return to the Printer Model selection dialog and chose the correct Printer Model and PPD combination there.

PostScript Options

PostScript specific options that are not catered for by the PPD.

- **Rotate landscape pages** checkbox: For certain printers, printing in landscape orientation requires rotating the page content. Typically this cannot be derived from the PPD, nor does the PPD contain instructions for carrying out this rotation.

In such a scenario, selecting this checkbox will have those printers print in Landscape mode.

Rules

This group displays the print options for the template along with their associated triggers, as well at the more general Simplex/Duplex options.

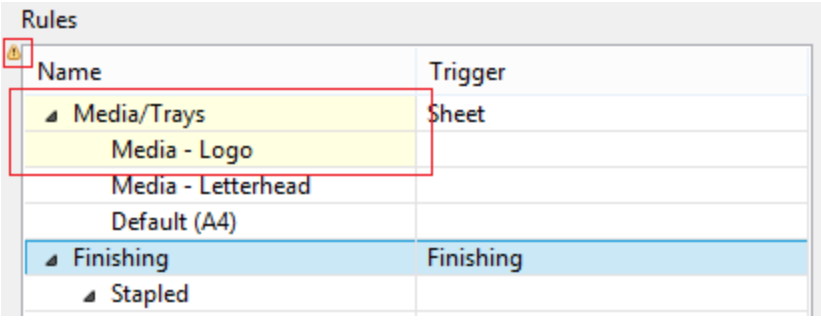
If your Template already has **Finishing** and **Media/Trays** entries set, then those options will appear pre-populated in the Rules table along with their Triggers, to allow you to assign specific options as **Printer Option** entries.

Additional Rules (and Triggers) can be added (or deleted) and all Rules can be edited, and/or moved up or down within the table after being added.

Within a group of Rules, the Rules are processed sequentially, from top to bottom. Thus if two Rules apply to the same option (such as two different Rules applying for Tray 1), then the topmost Rule will be applied, once the Rule trigger is encountered. Thus it makes sense to have the most specific rules at the top of a group, with the more generic rules placed below them.

Note

Rules should have at least one **Printer Option** assigned to them. If they do not, then the page is considered incomplete and a warning symbol will be displayed and the offending Rules highlighted. As seen in the following screen-shot:



If a **Media/Trays** Rule has an associated **Printer Option** that matches a Media entry in the Template or Presets, then it is highlighted in light green.

It is possible to create a set of Rules in which none of the Rules ever evaluate to true, and thus no Rule will be applied. This could be done on purpose, to allow the printer's default settings to apply.

The types of Rules are as follows:

- **JCL** (Job Control Language) Rules: This option only appears if the PPD contains a "`*JCL keywords`" section.
  Certain printers require Printer Job Language (a type of JCL) commands to function properly, or to allow certain features.
  If such entries are included in the PPD, then rules can be added to use them via the **Add** button.
  NOTE: JCL is a generic way to refer to languages like PCL's Printer Job Language.
- **Prolog** Rules: This adds options to the Prolog section of a PostScript document.
  You can add Prolog settings via the **Add** button.
- **Setup** Rules: This supplies PostScript resources for document or page setup.
  You can add Setup settings via the **Add** button.
- **Media/Trays** Rules: This allows you to set Media/Trays options. The Media/Trays settings will be pre-populated from settings contained within the current Connect Template, if such exist.
  They can also be added from other Templates, selected using the **Import** button.

  You can import settings from as many Templates as you like.
  You can also add additional Media/Trays settings via the **Add** button.

  Media/Tray selections always apply to *Sheets*.

  Once the Media/Trays are entered you will need to assign specific Printer Options (from those contained in the PPD) to the individual Media/Tray types.
- **Finishing** Rules: This allows you to set Finishing options.
  These settings will be pre-populated from settings contained within the current Connect Template, if such exist.
  They can also be added from Job Creation Presets and other Templates, selected using the **Import** button.

  You can import settings from as many Templates as you like.
  You can also add additional Finishing settings via the **Add** button.

  The Finishing Rules support all industry standard binding options.

Once the Finishing Rules are entered you will need to assign specific Printer Options (from those contained in the PPD) to each of the individual Finishing Rules.

> **Note**
>
> You should also choose an Option to switch off the Finishing selection, to cater for subset Finishing.

- **Plex** rules: This fixed group allows you to set rules for sheet plexing. The choices are Duplex, Simplex or Tumble Duplex.
  You should add Printer Options for each of these Plexing Rules.
- **Custom** rules: The Custom group does not appear by default, but can be added at any point via the **Add** button.

The **Rules** Options. These options can either be selected via the icons to the right of the Rules table, or from right click context sensitive menus within the Rules table.
The options are:

- **Import**: Allows importation of Rules from existing Job Creation Presets and/or Connect Templates. You will be presented with options to either **Import Finishing Settings**, or **Import Media Settings**, based upon what is currently selected and/or available in the Rules table.
  Both options allow you to select an existing Connect Template, from which to import the settings from, whilst Finishing settings can also be imported from Job Creation Presets. The settings will be added to those already in the Rules table.
- **Add**: This allows you to Add various options, dynamically.

  What options are available is dependant upon what rule is currently selected. For example, the *Plex* group does not have an option to **Add Rules** available to it.

  1. **Add Rule**: This will add a Rule that is specific to the type of option was selected when the Add Rule was pressed. The options are as follows:
     - **Prolog** : If the Rule is being added to a *Prolog* selection, then the "Rule name" on page 798 will be launched.You should add Printer Options for each of the Rules added. See the "Printer Options" on page 795 section for how to do this.

- **Setup** : If the Rule is being added to a *Setup* selection, then the "Rule name" on page 798 will be launched.
  You should add Printer Options for each of the Rules added.See the "Printer Options" on page 795 section for how to do this.
- **Media/Trays** : If the Rule is being added to a *Media/Trays* selection, then the "Dynamic Tray Mapping Editor Dialog" on page 800 will be launched. This dialog allows the description of standard Media options, such as paper Weight and Color.
  You should add Printer Options for each of the Rules added.See the "Printer Options" on page 795 section for how to do this.
- **Finishing** : If the Rule is being added to a *Finishing* selection, then the Dynamic Finishing Editor Dialog will be launched.
  You should add Printer Options for each of the Rules added.See the "Printer Options" on page 795 section for how to do this.
- **Plex** : Add Rule is not available as an option for *Plex* settings.
  You should add Printer Options for each of the Rules added.See the "Printer Options" on page 795 section for how to do this.
- **Custom** : If the Rule is being added to a *Custom Group* selection, then the "Rule name" on page 798 will be launched. The *Custom Group* selection allows for considerable customization and complexity.
  For example, they could be used to separate large documents (that might not fit into an Inserter machine), or to use a letterhead stock that is based upon what city the document recipient lives in.
  You should add Printer Options for each of the Rules added.See the "Printer Options" on page 795 section for how to do this.

2. **Add Custom Group**: This launches the "Custom Group Editor Dialog" on page 802 which adds a new Custom Group entry, and associated Trigger.
3. **Add PSHeader Group**: This adds a PostScript Header group (triggered on File).
   To add the actual Rule details you must select the new Header entry and then select the **Add Rule** option.
   Note that these rule are added in order the appear.
4. **Add Media/Trays Group**: This adds a new blank Media/Trays entry.
   To add the actual Rule details you must select the new Media/Tray entry and then select the **Add Rule** option.

5. **Add Finishing Group**: This adds a new blank Finishing entry.
   To add the actual Rule details you must select the new Finishing entry and then select the **Add Rule** option.

Each selection allows you to set the appropriate details for that type of rule.

- **Edit**: This allows you to edit the current Rule, dynamically. What options are available is dependant upon what rule is currently selected.
  The options are as follows:
  - **Media/Trays** : If the Rule is being edited is a *Media/Trays* selection, then the "Dynamic Tray Mapping Editor Dialog" on page 800 will be launched.
  - **Finishing** : If the Rule is being edited is a *Finishing* selection, then the Dynamic Finishing Editor Dialog will be launched.
  - **Plex** : *Plex* Rules cannot be edited.
  - **Custom / Prolog / Setup** : If the Rule being edited is any of these, then the "Custom Group Editor Dialog" on page 802 will be launched.

  Rules can also be edited via double mouse clicking on the rule. This launches the appropriate Editor Dialog, as listed above.
- / **Move Up / Down**: Move the selected rule(s) up or down within the Rules table. This allows sorting of options, for easier legibility, and more importantly, *for Order Dependency*.
  Within a group of Rules, the Rules are processed sequentially, from top to bottom. Thus if two Rules apply to the same option (such as two different Rules applying for Tray 1), then the topmost Rule will be applied, once the Rule trigger is encountered. Thus it makes sense to have the most specific rules at the top of a group, with the more generic rules placed below them.
- **Delete**: Delete the selected rule(s) from the Rules table.

Rule Details

This table displays the details of the currently selected Rule.

For Media/Trays, Finishing and Plex selections only the **Properties** are displayed. For Custom selections, the **Condition** can also be displayed.

The tabs can be broken down as follows:

- **Properties** Tab: displaying all the relevant Rule Details. These are:
  - **Property**: The list of properties available to this Rule.
    What properties are displayed is dependent upon the type of Rule selected.

- **Value**: The property's associated value, if set. If not set, the Value will remain blank.

The table is read only and the details cannot be adjusted or edited from within the table. Rules need to be edited from within the Rules table directly, through either the Edit button or by double mouse clicking on the Rule.

- **Condition** Tab: displays the logical condition that needs to be fulfilled before the Custom Rule is to be applied.

Printer Options

This section allows for the selection and ordering of Printer Options to be associated with the active Rule. The Print Options are all those contained in the PPD.

If no Rule is active, the Printer Options section will not be available.

The **Printer Options** table displays :

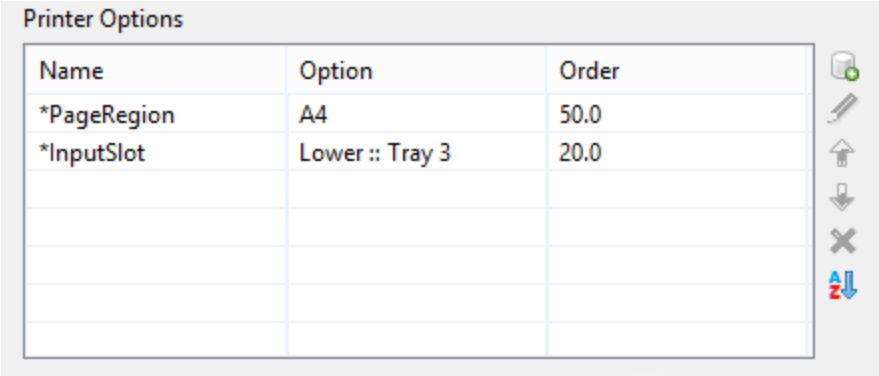
- **Name** column: This column contains the names of the Printer Options, as taken from the PPD file.
- **Option** column: This column shows the selected Printer Option preference.
- **Order** column: This column displays any Dependency Order criteria that applies to the selected Printer Option.

The choices available to the Printer Options are as follows:

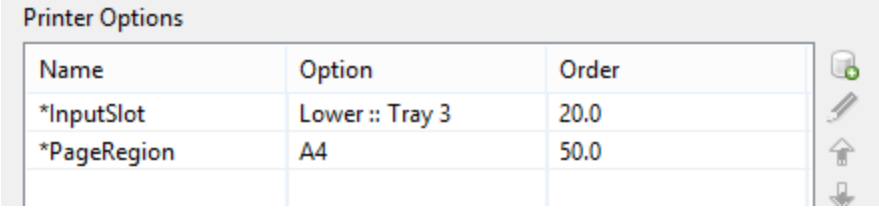
- **Add**: This launches the "Add Printer Option dialog" on page 802 which allows you to add one or more Printer Options. That dialog will be populated with options that relate to the Rule that was selected, when the Add Printer Options button was pressed.
- **Add custom PostScript**: This launches the "Add Custom Printer Option dialog" on page 806 which allows you to add customized PostScript instructions.
  Note that this option requires an in depth understanding of both the PostScript language and the specific printer itself.
- **Edit**: This allows you to edit an existing Rule Printer Option. It launches the "Add Printer Option dialog" on page 802.
- / **Move Up / Down**: Move the selected Printer Option(s) up or down within the table. This allows sorting of options for easier legibility, but manually moving options up and down is done without any check on Order Dependency.

- **Delete**: Delete the selected Printer Option(s).
- **Sort by Order dependency**: This sorts the Printer Options by the order of their dependencies.
  Not all Printer Options have dependencies, but for those that do, the order is important. For example, consider these following PPD file Printer Option extracts:
  - ***PageRegion:***
    ```
    *OpenUI *PageRegion: PickOne
    *OrderDependency: 50 AnySetup *PageRegion
    ...
    *CloseUI: *PageRegion
    ```
  - ***InputSlot:***
    ```
    *OpenUI *InputSlot: PickOne
    *OrderDependency: 20 AnySetup *InputSlot
    ...
    *CloseUI: *InputSlot
    ```

  We can see that one has an Order Dependency of 50 and the other 20.
  If *PageRegion* was selected prior to *InputSlot* in the "Add Printer Option dialog" on page 802, then they would appear like this in the Print Wizard:



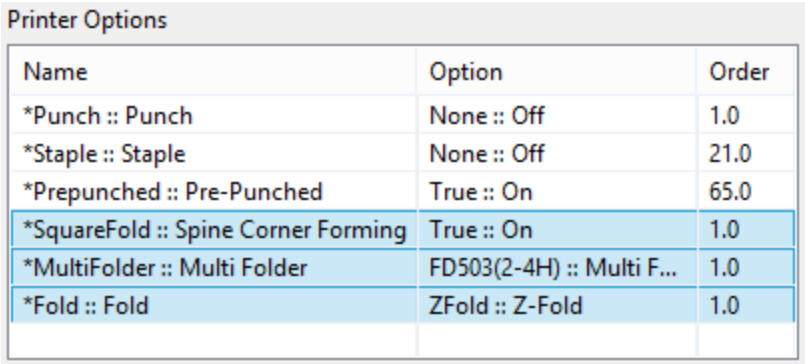
To sort them into the correct dependance order, use the **Sort by Order dependency** button. This will sort them into the proper dependency sequence. As seen below, where the lower dependency now appears first.



Warning

Nothing prevents the selection of conflicting and contradictory Print Options.

This can be seen in the following example, which shows multiple conflicting Folding Options have been chosen:



The breadth and diversity of PPD files as well as individual printer capabilities mean that the Print Wizard cannot meaningfully validate the selections made, and thus does not attempt to.

So you must be careful when making selections. We recommend that you always test each individual selection made, to confirm that they are valid and that they do what you expected of them.

This could entail considerable trial and error, but should lead to less anguish in the long run.

Advanced Print Wizard and Output Creation Preset Wizard navigation options

- **Preview** button (active in ***Advanced Print*** Wizard only): Click to launch a Proof Preview window, which displays how the printed output would look based upon the currently

chosen selections.

- **Back** and **Next** buttons: Used to navigate back and forth through all the selected options within the Wizard. Up until the Print button is pressed, one can reverse all the way through the wizard to return to the main selection page (the "Print options" on page 704 page) and add or remove printing options from the print run.
- **Print** button (***Advanced Print*** Wizard only) or **Finish** button (***Output Creation Preset*** Wizard only):
  Click to produce print output/finalize the Preset according to the current settings.
  This can be done at any point within the Wizard, whether or not *all* the options selected in the "Print options" on page 704 page have been completed or not.
- **Cancel** button: Cancels the Print Wizard, without creating any printout.

Dynamic Custom Element Editor Dialog

Use this dialog to create Rules for determining whether a Custom setting in the "Dynamic PPD Options" on page 788 page of the Printer Wizard should be applied or not.

The **Dynamic Custom Element Editor** dialog is basically a logical Rule creation dialog.

It can be broken up as follows:

Rule name

Either enter or update the Custom Rule name here. The "Dynamic PPD Options" on page 788 page will be updated with the choice thereafter.

Group

A Group consists of one or more sub-Group, Size, Position or Metadata rules with a logic operator.

There may be one rule or many rules at the same level, and there may be Groups nested within Groups This provides the ability to create quite complex nested logical structures.

Four logic choices are available at the Group level. These are:

- **All of the following**.
  This equates to the logical operator (... AND ...)

If *all* of the following criteria are met, then this group resolves to TRUE.

- **Any of the following**.
  This equates to the logical operator (... OR ...)
  If *any* of the following criteria are met, then this group resolves to TRUE.
- **Not all of the following**.
  This equates to the logical operator (NOT(... AND ...)).
  If *any (but not all)* of the following criteria are met, then this group resolves to TRUE.
- **Not any of the following**.
  This equates to the logical operator (NOT (... OR ...))
  If *none* of the following criteria are met, then this group resolves to TRUE.

The top level is always a Group. A Group can contain one or more sub-Groups.

Rule

A Rule is a logic expression using a single operator or function, and their associated operand (s). The number and type of operands is dependent upon the operator or function chosen.

The Rule types are as follows:

- **Group**:
  A single top level Group, plus nested sub-Groups. This nesting of Groups allows for nesting of logical operations.
  For Prolog or Setup entries this is the only option available. In those cases it is to be used so as to set the Rules to always be TRUE, and thus always apply.
- **Position** (not available to Prolog or Setup entries):
  This rule operates upon the position of the selected group within the broader group. It operates at *Page, Sheet, Document, Document Set* and *Job Segment* levels.
  The logical operators that apply a totally dependent upon what grouping the position is to apply to.
- **Size** (not available to Prolog or Setup entries):
  This rule operates upon the size of the selected group. This could be the amount of sub-groups contained within, or the amount of physical sheets. The logical operators that apply a dependent upon the group the size test is being applied to.
- **Metadata** (not available to Prolog or Setup entries):
  This rule operates upon Metadata entries. Select the Metadata Field, then select the logical test to apply to it, along with the associated test value(s).
  The Metadata options available are all text based comparisons. They compare whether

the Metadata field "*is equal to*", "*is not equal to*", "*contains*" or "*does not contain*" the selected text value.
Case sensitivity can be added or removed from the comparison, via the **Case sensitive** check-box.

> **Note**
>
> Metadata will only appear as a rule option if Metadata is available. Which means that the metadata option needs to have been selected and set within the Print Wizard.

Expand button

To allow logical constructs than are not currently catered for by the **Dynamic Custom Element Editor**, the **Expand** button is available at any time.
The Expand button translates all currently configured rules into JavaScript, which can then be manually edited and expanded.

An example of where this might be useful is in converting a text comparison to a case insensitive comparison. To do so, click the Expand button to convert the rule into JavaScript, then add `.toLowerCase()` to the relevant sections, making them case insensitive.

> **Warning**
>
> The Expansion is a one-way translation that cannot be undone. Once expanded, the rule stays expanded.
>
> The only way to go back to the previously configured rule(s), is to **Cancel** the Rule editor dialog.

Dynamic Tray Mapping Editor Dialog

The **Dynamic Tray Mapping Editor** dialog allows you prepare a Rule for the associated **Media/Tray** group in the "Dynamic PPD Options" on page 788 page and for the **Mapped PPD Entries** group in the "PPML Options" on page 726 page of the Printer Wizard.

The **Dynamic Tray Mapping Editor** dialog can be broken up as follows:

- **Rule name** edit box: This displays the previously assigned Rule name, or the default Rule name, which is Media/Tray**n**, where **n** is a numeric sequence, starting at 1 for the first rule, and incrementing for any subsequent rule(s).
  The Rule name can be changed in this edit box, if desired.
- **Properties** group: This section contains all the properties that can be set for the Media/Trays selection.

  The properties are optional, but for the **Size** and associated **Height** and **Width** properties. The exception is the "*<<Not Set>>*" **Size** selection. If this is chosen, then the **Size**, **Height**, **Width** and **Orientation** values are all ignored.

  The properties include:

  - **Name** edit box: Here you can assign a secondary name to the Rule. Any name entered here will be appended within brackets to the greater **Rule name**.
    For example, if you have a **Rule name** of "*Default*" and you entered "*A4*" here, then the name would appear as "*Default (A4)*" in the "Dynamic PPD Options" on page 788 page of the Printer Wizard.
  - **Color** drop down list box: Used for selecting the color of the stationery that should be present in the tray.
  - **Type** drop down list box: Chose the stationery type from the selection of industry standard options available in the drop down box, or choose *Unspecified* to leave undefined.
  - **Weight** edit box: Add the stationery paper weight, if desired. The weight in grams, and must be entered as a number.
  - **Size** drop down list box: Select the paper size here. Either select from one of the industry standard stationery size options, or chose *Custom* to manually set the stationery dimensions as desired.
  - **Width** and **Height** edit boxes. These display the dimensions if an industry standard paper **Size** were selected, and do not allow modification. These boxes will only allow editing if a *Custom* paper **Size** were chosen.
  - **Orientation** selection: Choose between *Portrait* or *Landscape*.
    Selecting *Landscape* will reverse the current **Height** and **Width** values and convert the paper **Size** to *Custom*.
  - **Optimize Media Calls** check box: Select if consecutive records are likely to use the same output media settings. This setting allows those consecutive records to share

media details rather than recreating them for each page, reducing print output size and improving throughput.

Custom Group Editor Dialog

The **Custom Group Editor** dialog allows you associate a Custom Rule with a trigger. It is launched when either adding a new Custom Group rule, or editing an existing one in the "Dynamic PPD Options" on page 788 page of the Printer Wizard.

The **Custom Group Editor** dialog can be broken up as follows:

- **Group name** edit box: What name to give the Custom Group Rule in the "Dynamic PPD Options" on page 788 page.
- **Trigger on** drop down list box: Select from a series of available Connect Template resource related triggers.

Add Printer Option dialog

The **Add Printer Option** dialog allows you to add a specific PostScript printer option, from those included in the printer's PPD file.

The dialog is launched from the Print Wizard "Dynamic PPD Options" on page 788 or the "PPML Options" on page 726 page.

The **Add Printer Option** dialog can be broken up into 3 distinct groupings, **Sections**, **Options** and **PostScript Instruction**.

Sections

In PPD files all options are grouped into Sections. Each Section typically represents a particular logical function, such as stapling, or binding. However, a PPD can contain many Sections and many may not be at all relevant to Finishing. By default, the dialog displays just those Sections which are likely to be the most relevant.

Choose what Section you want from the Section selection box. The sections can be sorted or filtered as follows:

- **Display all Sections** check-box: This will have the Section selection box display *all* the PPD Sections. By default it is turned off, as the **Add Printer Dialog** automatically limits the Sections to those which are most likely to be relevant.
  If you cannot find the Section you want in the initial display, however, un-check this

check-box to display *all* the PPD Sections.

- **Filter Sections** edit box: This will filter the displayed Sections based upon whatever is entered in the edit box.
  For example, an entry of "Pa" would display "PageSize" and "PageRegion", but not "Duplex", "Collate", or any other entry that does not contain the string "pa".

  > **Note**
  >
  > The filter is case insensitive, and applies to the entire string, not just the first few characters.

- Sort: Sections are displayed in the order in which they appear in the PPD. Use this button to sort the entries in alphabetic order, or to revert back to the original PPD order.

Options

Choose what Option to apply from those available in the Options selection box. The Options presented in the selection box are all those that apply to the Section chosen in the Section selection box.

- **Filter Options** edit box: This will filter the displayed Section Options, and is based upon whatever is entered in the filter edit box.
  For example, filtering Media on "a", will return "**A**3", "**A**4", "Leg**a**l" and "T**a**bloid" but not "Letter", or any other entry that does not feature the character "a".

  > **Note**
  >
  > The filter is case insensitive, and applies to the entire string, not just the first few characters.

- Sort: Options are displayed in the order in which they appear in the PPD. Use this button to sort the entries in alphabetic order, or to revert back to the original PPD order.

PostScript Instruction

This is the PostScript Instruction that is created from the selection of Section and Option. If there is no PostScript Instruction present, even though a Printer Option has been selected, then that

Printer Option is not actually implemented. This will be the case with some manufacturers PPD files.

Usage example

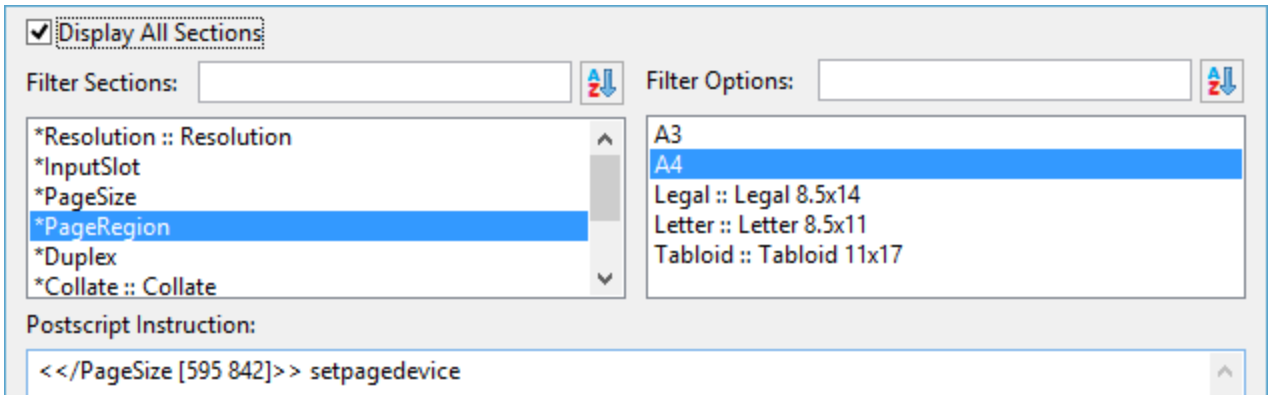
By way of example on how to use the dialog, please see the following PPD extract.

This is the "PageRegion" Section for the Oce 3165PS GB printer:

```
*OpenUI *PageRegion: PickOne
*OrderDependency: 50 AnySetup *PageRegion
*DefaultPageRegion: A4
*PageRegion A3: "<</PageSize [842 1191]>> setpagedevice"
*PageRegion A4: "<</PageSize [595 842]>> setpagedevice"
*PageRegion Legal/Legal 8.5x14: "<</PageSize [612 1008]>> setpagedevice"
*PageRegion Letter/Letter 8.5x11: "<</PageSize [612 792]>> setpagedevice"
*PageRegion Tabloid/Tabloid 11x17: "<</PageSize [792 1224]>> setpagedevice"
*CloseUI: *PageRegion
```

The PPD provides 5 page options: A3, A4, Legal, Letter and Tabloid, with A4 as the default.

After selecting "*PageRegion" in the Section selection box, the associated options are presented in the same order as within the PPD, with the default PPD option (A4) being the active selection.
As seen in the following dialog extract:



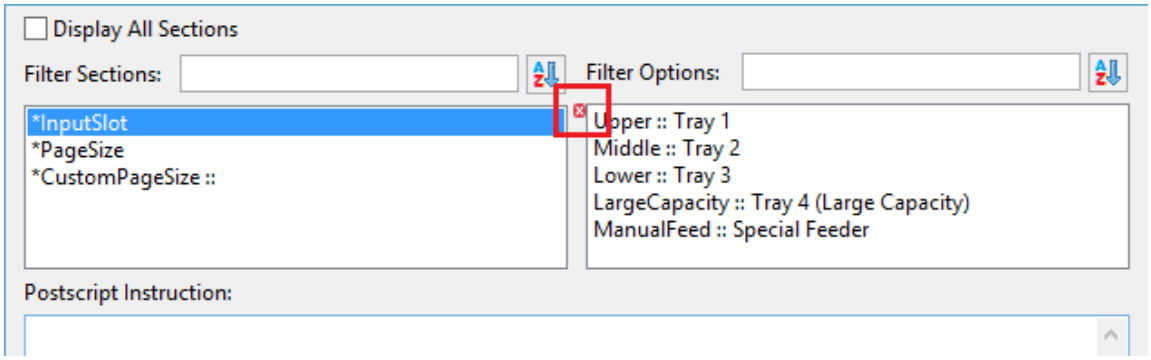
The next PPD extract relates to the "InputSlot" options available to the same Oce 3165PS GB printer.

```
*OpenUI *InputSlot: PickOne
*OrderDependency: 20 AnySetup *InputSlot
*DefaultInputSlot: Unknown
*InputSlot Upper/Tray 1: "<</InputAttributes << /Priority [0] >> >> setpagedevice"
```

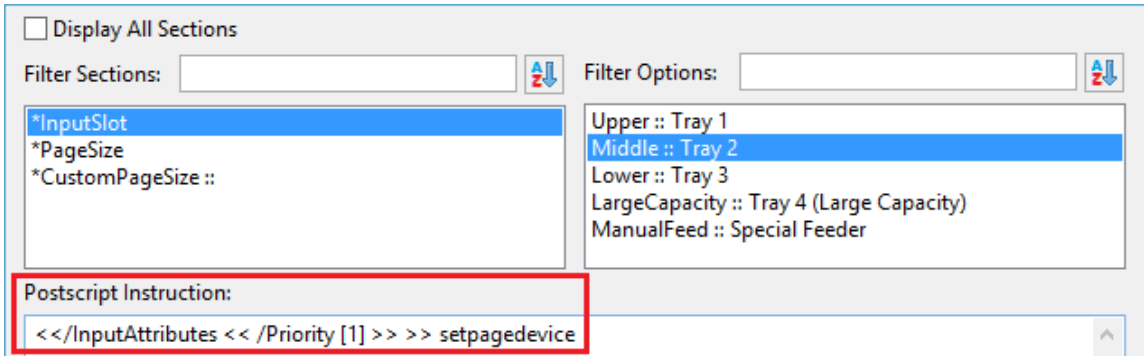
```
*InputSlot Middle/Tray 2: "<</InputAttributes << /Priority [1] >> >> setpagedevice"
*InputSlot Lower/Tray 3: "<</InputAttributes << /Priority [2] >> >> setpagedevice"
*InputSlot LargeCapacity/Tray 4 (Large Capacity): "<</InputAttributes << /Priority
[3] >> >> setpagedevice"
*InputSlot ManualFeed/Special Feeder: "1 dict dup /ManualFeed true put
setpagedevice"
...
*CloseUI: *InputSlot
```

It provides 5 tray options, but with no default selection.

As there is no default "InputSlot" selection, the Options selection box will display all the options as well as a red cross to indicate that there is an issue that needs to be addressed.
As seen in the following dialog extract:



Once an option is selected, then the red cross is removed and the PostScript Instruction is completed. As seen here:



If there is no PostScript Instruction present, even though a valid Printer Option has been selected, then that Printer Option has not been actually implemented in the PPD. This will be the case with some manufacturer PPD files.

Add Custom Printer Option dialog

The dialog is launched from the Print Wizard "Dynamic PPD Options" on page 788 page.

The **Add Custom Printer Option** dialog allows you to add your own customized PostScript printer snippets. The custom PostScript can utilize dynamic Connect metadata variables that will get resolved when the custom PostScript is run.

> **Note**
>
> PostScript coding can be an arcane art, given the wildly differing capabilities of PostScript printers.
> We recommend that only those fully conversant with both their printer and the PostScript language itself attempt to write complex customized PostScript code.

Usage Scenarios

Tailored PostScript snippets could be used in a number of ways. Such as:

Selecting a custom paper size

Sometimes a printer supports a custom paper size, but requires that paper size to be explicitly set. This cannot be done with a printer option from a PPD, because a PPD does not have a mechanism for variables.

A custom PostScript snippet could be used to enable this functionality.

Selecting the number of copies (NumCopies) to print

OL Connect currently does not currently support setting the number of copies to print on a printer that itself can create copies. To currently achieve such functionality OL Connect needs to create multiple copies of the content in the print file itself, and have each sent to the printer.

However the new custom PostScript functionality could allow the printer to print multiple copies of the one job, via the PostSript `NumCopies` instruction. This instruction could be combined with runtime variables (metatdata) to dynamically select the number of copies to print.

The following Usage example shows how to do this.

Usage example

The following PostScript instruction determines the number of copies to be printed. The amount of copies is contained in the metadata "Copies" entry, which will be set at runtime.

```
%%BeginFeature:
<</NumCopies ${Number(document.metadata.Copies),0.###,en} >>
setpagedevice
%%EndFeature
```

Dialog Interface

The dialog consists of the following options:

- **Name**: Enter a name for the customized .
- **Custom PostScript code**: Enter your PostScript code here. The code can incorporate document metadata, which is accessible via the Add button.

  A reminder that PostScript code is case sensitive.

Separation options

This page appears as part of the **Advanced Print Wizard** and the Output Creation Preset Wizard.

The **Separation Options** page defines how to separate a job into discreet portions, using either predetermined counts (of *Sheets*, *Documents*, *Document Sets* or *Job Segments*), slip sheets, or jogging.

- **Separate by Count** group:
  This group allows for the splitting of output based upon a specified number of *Sheets*, *Documents*, *Document Sets* or *Job Segments*.
  - **Split**: Use the drop-down to select whether or not to split the job based on number of *Sheets*, *Documents*, *Document Sets* or *Job Segments*.

        - **None**: Select this option to ignore count splitting.
        - **At Exactly**: Select to create a split based upon a predetermined number of *Sheets*, *Documents*, *Document Sets* or *Job Segments*.
    - **Every**: Enter the number of *Sheets*, *Documents*, *Document Sets* or *Job Segments* at which the output is to be split.
      This option is only available if the **Split** entry has been set to *"At Exactly"*.
    - Use the drop-down box to chose what the separation option is to be.
      The selection is either **Sheets**, **Documents**, **Document Sets** or **Job Segments**.
      The default separation option is *Sheets*, which was the only option available for splitting prior to Connect 2019.1.
- **Separation Settings** group:
  This setting is only available if no **Separate by Count** split has been specified.
    - **Separation**: Use the drop-down to select when a job separation occurs, which is either **None** (no separation) or at the **Job**, **Job Segment**, **Document** or **Document Set** level.
- **Slip Sheets** group:
    - **Add slip sheet**: Use the drop-down to select whether to add a slip sheet before or after a specific separation, or whether to use none.
    - **Every**: Use the drop-down to select at which separation to add a slip sheet, at the **Job**, **Job Segment**, **Document** or **Document Set** level.
    - **Media Size**: Use the drop-down to select the media size of the slip sheet.
      If a custom Media Size was chosen:
        - **Width**: enter slip sheet page width.
        - **Height**: enter slip sheet page height.
- **Jog** group:
    - **Jog after every**: Use the drop-down to select when to jog the printer, which is either **None** (no forced jogging) or at the **Job**, **Job Segment**, **Document** or **Document Set** level.

Advanced Print Wizard and Output Creation Preset Wizard navigation options

- **Preview** button (active in ***Advanced Print*** Wizard only): Click to launch a Proof Preview window, which displays how the printed output would look based upon the currently chosen selections.

- **Back** and **Next** buttons: Used to navigate back and forth through all the selected options within the Wizard. Up until the Print button is pressed, one can reverse all the way through the wizard to return to the main selection page (the "Print options" on page 704 page) and add or remove printing options from the print run.
- **Print** button (***Advanced Print*** Wizard only) or **Finish** button (***Output Creation Preset*** Wizard only):
  Click to produce print output/finalize the Preset according to the current settings.
  This can be done at any point within the Wizard, whether or not *all* the options selected in the "Print options" on page 704 page have been completed or not.
- **Cancel** button: Cancels the Print Wizard, without creating any printout.

Designer Script API

In Designer templates, every bit of information can be tailor-made, using scripts. When Connect generates actual output – letters, web pages or emails -, it opens a record set and merges it with the template. It takes each record, one by one, and runs all scripts for it (in a specific order, see "The script flow: when scripts run" on page 413).

Most scripts can be made using one of the Script Wizards (see "Personalizing content" on page 338).
However, when you want to do more than what you can do with a Wizard, you may write a script yourself. If you are not familiar with writing scripts, please read "Writing your own scripts" on page 393 first.

All scripts in the Designer have to be written in JavaScript.
If you don't know JavaScript, the many examples given in this API will help you get started.
It is worth the effort, however, to familiarize yourself with the JavaScript syntax. For a simple script all you need to know can be found on the following web pages:
https://www.w3schools.com/js/js_syntax.asp and https://www.w3schools.com/js/js_if_else.asp.

In the editor window, press **Ctrl** + **Space** to see the available features and their descriptions.
Use the arrow keys to select a function or object and press Enter to insert it in the script.
Type a **dot** after the name of the function or object and press Ctrl + space again to see which features are subsequently available.
For more keyboard shortcuts, see "Keyboard shortcuts" on page 561.

Designer API

The "Standard Script API" below describes the objects and functions that are available in **template scripts**, created inside the Scripts pane. Template scripts change the contents of sections in a template.

> **Note**
>
> In a Print context, the scripts in the Scripts pane run once for each section and once for each Master Page (see "Master Pages" on page 156).

Control Script API

Control Scripts are a special kind of Designer Scripts. They don't touch the content of the sections themselves, but they change the way a template is outputted, for example by selecting or omitting sections from the output.
For more information about Control Scripts and their use, see "Control Scripts" on page 429.
Features that are specific to Control Scripts are listed in the "Control Script API" on page 931.

Post Pagination Script API

Post Pagination Scripts are run in a **Print** context **after** the content has been paginated. Because they can search through the output of all Print sections, and modify Print sections (one at a time), they may be used to create a Table Of Contents (TOC), as explained in the topic: "Creating a Table Of Contents" on page 447.

For more information about Post Pagination Scripts and their use, see "Post Pagination Scripts" on page 445.
Features that are specific to Control Scripts are listed in the "Post Pagination Script API" on page 967.

Standard Script API

This page lists the global objects and functions that are available in Standard Scripts, created inside the Scripts pane. Click through to an object or function to get a description and examples.

Most of these objects and functions are also available in Control Scripts ("Control Scripts" on page 429) and Post Pagination Scripts ("Post Pagination Scripts" on page 445).
For objects and functions restricted to Control Scripts see "Control Script API" on page 931, and for the Post Pagination Script API see "Post Pagination Script API" on page 967.

The basics of script-writing in the Designer are explained in the following topic: "Writing your own scripts" on page 393.

Objects

Object	Description
"results" on page 972	This object - of the type: QueryResults - is used to manipulate the content of the template. It contains the HTML element or set of HTML elements that match the selector of the script, specified in the script editor. This object is not available in Control Scripts, because that type of script doesn't have a selector (see "Control Scripts" on page 429).
"this" on page 891	A Standard script or Post Pagination script that has its scope set to "Each matched element" (see "Setting the scope of a script" on page 399) will be called in a loop over the elements that match the selector of the script - the `results` (see "results" on page 972). In such a script, `this` is an object of the type QueryResult. It represents the current element in the loop. This object is not available in Control Scripts, because that type of script doesn't have a selector (see "Control Scripts" on page 429).
"record" on page 848	The record in the main data set that is currently being merged. To get the value of a field in the record, use `record.fields['fieldname']` or `record.fields.fieldname`.
"logger" on page 832	Global object that allows you to log messages.
locale	Defines which locale to use. (See "Locale" on page 336.) Note that the value is read-only.

Object	Description
"formatter" on page 817	Global object that allows you to format values (such as a date or number).
automation	This object encapsulates the properties of the Workflow process that triggered the current operation. Not available in PrintShopMail Connect.
"merge" on page 989	The `merge` object is mainly used in Control Scripts. It gives access to the template with all of its contexts, sections and runtime parameters. It doesn't give access to the content of the sections. To change the content of a section, you would create a script with a selector and use the `results` object in the script (see "results" on page 972).
"contentitem" on page 968	(Print output only.) The `contentitem` object holds the Print Content Item that is written to the Connect database when generating Print output. It allows you to add custom properties to the Print Content Item, in the form of key-value pairs (a JSON string). **Note**: The `contentitem` object is not available in PrintShopMail Connect, due to the absence of a Connect database.

Global functions

Function	Description
"fatalError (message)" on page 815	Triggers a fatal error that aborts content creation.
"loadhtml()" on page 824	Loads HTML data from an HTML (snippet). The returned HTML can be placed into a variable or into a set of HTML elements.
"loadjson()" on page 827	Loads JSON data from a URL. This is a simple way to retrieve content from external systems.
"query()" on	Performs a query in the template's contents and creates a new result set

Function	Description
page 845	containing the HTML elements that match the given CSS selector.
"resource()" on page 940	This function returns information about an image resource. It can also be used to check if a file exists.
"translate()" on page 930	Gets the translation of a source text with an (optional) context. It returns the message as-is if no translation is found.

Examples of iterator functions

Function	Description
"each()" below	A generic iterator function, to iterate over the elements in the result set.
"for(... in ...)" on page 816	Iterates over the enumerable properties of an object, in arbitrary order. For each distinct property, statements can be executed.

each()

A generic iterator function, to iterate over the elements in a result set.

> **Tip**
>
> Instead of using the `each()` function to loop over a result set, you could set the scope of a Standard script or Post Pagination script to "Each matched element" and access each matched element directly via the `this` object. See "Setting the scope of a script" on page 399 and "this" on page 891.

each(callback)

Iterates over the elements in a set, such as the enumerable properties of an object, in arbitrary order. For each distinct property, statements can be executed.

callback

A function. The callback function is passed the iteration index and the current element. In the scope of the callback function, `this` refers to the current element.

Examples

The following scripts demonstrate a simple iteration over the elements in the `results` (the set of HTML elements that match the selector of the script).

The first script sets the background color of each of the elements to red. (This is just to demonstrate how this function works. It is easier to change the style of a set of HTML elements using the `css()` function; see "css()" on page 909.)

```
results.each(function(index){
        results[index].css('background-color','red');
});
```

This script adds a random integer to each element in the result set.

```
results.each(function(index){
        var test = Math.floor((Math.random() * 10) + 1);
        this.html(test);
});
```

Selector	Matched element	Matched element after script execution
p	 	3 1 7

The next script gets the row index (of the current element in the set) and puts it in a paragraph.

```
results.each(function(index){
        this.text(index);
}
```

Selector	Matched element	Matched element after script execution
p		0

Selector	Matched element	Matched element after script execution
	 	1 2

Using `each()` in a translation script

The following script first loads a snippet containing translation strings, depending on the value of a field. Then it inserts translations by iterating over elements in the `results` (the set of HTML elements that match the selector of the script) and setting the HTML of each element with a value from the array of translation strings.

```
var strings = loadjson('snippets/' + record.fields.locale +
'.html');
results.each(function(index){
        if( strings[this.attr('data-translate')])
        this.html(strings[this.attr('data-translate')]);
});
```

Note: for documentation on the `data-*` attribute, see https://www.w3schools.com/tags/att_global_data.asp.

Selector	Matched element	Matched element after script execution
p	 	primero último dirección de correo electrónico

fatalError(message)

The `fatalError()` function triggers a fatal error that aborts content creation.

message

A string.

When a script calls this function in Preview mode, the script that triggers it is marked with an error icon in the Scripts pane, and the given message is displayed in a hint.

When generating output from the Designer, the Designer will log the error and display an error dialog with the given message. Content creation is aborted.

> **Note**
>
> Errors thrown in a `try` statement do not abort content creation. Content creation will continue with the next script, the same record.

for(... in ...)

Can be used to iterate over fields in a data set or rows in detail table. Also see https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/for...in.

for(`variable` in `object`) { ... }

Iterates over the enumerable properties of an object, in arbitrary order. For each distinct property, statements can be executed.

Examples

This script iterates over field names in the current record and adds them to a paragraph.

```
for(var i in record.fields){
        results.after("<p>" + i + "</p>");
}
```

Selector	Matched element	Matched element after script execution
#test	<h1 id="test">Fields</h1>	<h1 id="test">Fields</h1> first last email

This script iterates over fields in the current record, retrieving their values. Then it adds the values to a paragraph.

```
for(var i in record.fields){
        results.after("<p>" + record.fields[i] + "</p>");
}
```

Selector	Matched element	Matched element after script execution
#test	<h1 id="test">Fields</h1>	<h1 id="test">Fields</h1> Peter Parker pparker@localhost.com

formatter

The `formatter` is a global object that allows you to format values in a script.

The Text Script Wizard also allows you to format variable data; see "Using the Text Script Wizard" on page 353 and "Formatting variable data" on page 357.

> **Note**
>
> The TextFormatter object is now deprecated and will eventually be removed.

Functions

Function	Description
• currency() • currencyNoSymbol() • grouped() • integer() • integerUngrouped() • number() • numberUngrouped()	The `currency()`, `grouped()`, `integer()` and `number()` functions allow you to format a number, possibly with a custom pattern. See "Number functions" on page 843.
• date() • dateLong()	The `date()`, `dateTime()` and `time()` functions allow you to format a date and/or time in different ways. See "Date,

Function	Description
• dateMedium() • dateShort() • dateTime() • dateTimeLong() • dateTimeMedium() • dateTimeShort() • timeLong() • timeMedium() • timeShort()	date/time and time functions" on the facing page.
• lowerCase() • upperCase() • properCase()	The text formatting functions are used on Strings. `lowerCase()` transform all characters to lowercase, `upperCase()` transforms all characters to uppercase and `properCase()` transforms the first character of each word to uppercase and all other characters to lowercase.

Date, date/time and time functions

- date()
- dateLong()
- dateMedium()
- dateShort()
- dateTime()
- dateTimeLong()
- dateTimeMedium()
- dateTimeShort()
- time()
- timeLong()
- timeMedium()
- timeShort()

Note

The locale also influences the output of the different Date functions; see "Locale" on page 336.

date(value, pattern)

Formats a date object using a custom pattern.

value

A Date object. A Date can contain a date and time.

pattern

String. The custom pattern may consist of pattern letters, separating symbols and quoted text, for example: "MMMM dd, yyyy"; see "Date and time patterns" on page 822. Note that the repetition of pattern letters determines the exact presentation.

dateLong(value)

Formats a date as long string representation, for example **April 1, 2016**.

value

A Date object. A Date can contain a date and time.

dateMedium(value)

Formats a date as medium string representation, for example **01/04/16**.

value

A Date object. A Date can contain a date and time.

dateShort(value)

Formats a date as short string representation, for example **1-Apr-2016**.

value

A Date object. A Date can contain a date and time.

dateTime(value, pattern)

Formats a date and time object using a custom pattern.

value

A Date object. A Date can contain a date and time.

pattern

String. The custom pattern may consist of pattern letters, separating symbols and quoted text, for example: "yyyy.MM.dd G 'at' HH:mm:ss z"; see "Date and time patterns" on page 822. Note that the repetition of pattern letters determines the exact presentation.

dateTimeLong(value)

Formats a date and time as long string representation, for example **April 1, 2016 12:00:00 EDT AM**.

value

A Date object. A Date can contain a date and time.

dateTimeMedium(value)

Formats a date and time as medium string representation, for example **1-Apr-2016 12:00:00 AM**.

value

A Date object. A Date can contain a date and time.

dateTimeShort(value)

Formats a date and time as short string representation, for example **01/04/16 12:00 AM**.

value

A Date object. A Date can contain a date and time.

time(value, pattern)

Formats a time using a custom pattern.

value

A Date object. A Date can contain a date and time.

pattern

String. The custom pattern may consist of pattern letters, separating symbols and quoted text, for example: "'at' HH:mm:ss z"; see "Date and time patterns" on the facing page. Note that the repetition of pattern letters determines the exact presentation.

timeLong(value)

Formats a time as long string representation, for example **12:00:00 EDT AM**.

value

A Date object. A Date can contain a date and time.

timeMedium(value)

Formats a time as medium string representation, for example **12:00:00 AM**.

value

A Date object. A Date can contain a date and time.

timeShort(value)

Formats a time as short string representation, for example **12:00 AM**.

value

A Date object. A Date can contain a date and time.

Examples

Creating a Date object from a string

When you open a data file or database in the Designer, all fields are text fields (fields of the type `string`). The `formatter` cannot be used to format a string with a particular date format. The solution is to store the string in a variable as a Date object, and use the `formatter` with that variable.

The following sample script demonstrates this solution. It splits a string into parts and then creates a new Date object with the parts in the correct order. To construct a Date, the parts of the date must be put in the following order: year, month, day, and optionally hours, minutes, seconds, milliseconds (see https://www.w3schools.com/js/js_dates.asp and

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Date.)
When the time is omitted, it defaults to 12:00:00 AM.

```
/* Convert the string 21-12-1997 into a valid JavaScript date */
       var strDate = record.fields["date"];
       var dateParts = strDate.split("-");
       var date = new Date(dateParts[2], (dateParts[1] - 1), dateParts
[0]);
```

> **Note**
>
> JavaScript counts months from 0 to 11. January is 0. December is 11.

Another way to put a string in a Date is to use the `Date.parse` function; see https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Date/parse.

The `date` variable can be used as the value in the date, dateTime or time functions of the `formatter`.

```
var myDate = formatter.date(date, "MM/dd/yyyy");
```

The custom pattern that the script provides, outputs the month and day in two digits each and the year in four digits: 05/21/2016. For more examples of formatting patterns, see "Date and time patterns" below.

Date and time patterns

Dates and times in a template originating from a `date` field in a record set can be displayed using a custom pattern. In the Script Editor, the pattern can be passed to a date, dateTime or Time function of the `formatter`; see "formatter" on page 817.

The custom pattern may consist of pattern letters (see below), for example: "MM/dd/yyyy". The components can be separated with a space or a symbol, e.g. `.`, `/`, `-`. Text must be put in quotes.

The repetition of pattern letters determines the exact presentation. For example, if the number of pattern letters for a month is less than 3 (M or MM), the month is displayed as a number. If the number of pattern letters is 3 (MMM), it will be displayed as text; if available, a short or

abbreviated form of the month's name will be used. If the number of pattern letters is 4 or more (MMMM), the month's full name is displayed.

> **Note**
>
> The pattern letters and patterns on this page are only suitable for **displaying** dates and times in templates.

Pattern characters

Letter	Component	Presentation	Examples
G	Era designator	Text	AD
y	Year	Year	1996; 96
Y	Week year	Year	2009; 09
M	Month in year	Month	July; Jul; 07
w	Week in year	Number	27
W	Week in month	Number	2
D	Day in year	Number	189
d	Day in month	Number	10
F	Day of week in month	Number	2
E	Day name in week	Text	Tuesday; Tue
u	Day number of week (1 = Monday, ..., 7 = Sunday)	Number	1
a	Am/pm marker	Text	PM
H	Hour in day (0-23)	Number	0
k	Hour in day (1-24)	Number	24
K	Hour in am/pm (0-11)	Number	0
h	Hour in am/pm (1-12)	Number	12
m	Minute in hour	Number	30
s	Second in minute	Number	55
S	Millisecond	Number	978
z	Time zone	General time zone	Pacific Standard Time; PST; GMT-08:00
Z	Time zone	RFC 822 time	-0800

		zone	
X	Time zone	ISO 8601 time zone	-08; -0800; -08:00

Note

These date and time pattern letters and patterns conform to standard Java notation. For more information, see http://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html.

loadhtml()

Loads HTML content from the specified HTML file. The file may be located inside the template (see "Snippets" on page 285) or hosted on a Content Management System or on another location outside the template.

An optional `selector` allows you to retrieve only the content of matching elements.

Note

- The specified HTML file is expected to be UTF-8 encoded.
- Loadhtml() is cached per batch run (based on the URL) in print/email.

Tip

To load a JavaScript file (.js) or a style sheet (.css) you can use `loadtext()`. See "loadtext()" on page 829.

Tip

External content is not loaded while editing a script. To test a script that loads external content, you can do a preflight; see "Doing a Preflight" on page 405.

loadhtml(location)

Loads all HTML from the specified HTML file.

location

String containing a path that can be absolute or relative to the section/context.
Use `snippets/<snippet-name>` to retrieve the content from an HTML file residing in the Snippets folder on the Resources panel.
In order to retrieve files from outside the template the `file` protocol is supported as well as the `http/https` protocols.
The complete syntax of a fully qualified URL with the "file" protocol is: file://<host>/<path>. If the host is "localhost", it can be omitted, resulting in file:///<path>, for example: file:///c:/somefolder/somecontent.html.
When using the http/https protocol, remember that only absolute paths are supported inside remote snippets (see "Remote snippets" on page 286).

> **Tip**
>
> To quickly get the location of an HTML snippet in the template resources, right-click the snippet in the Resources pane and select **Copy Resource Location**.
> The path to a remote HTML snippet can be copied from the snippet's properties: right-click the snippet in the Resources pane and select **Properties**.

Examples

This script loads a local HTML snippet (from the Resources panel) directly into the matched elements

```
results.loadhtml("snippets/snippet.html");
```

The following script loads a local HTML snippet (Resources panel) into a variable. The replaceWith() command is used to replace the element(s) matched by the script's selector with the contents of the snippet.

```
var mysnippet = loadhtml('snippets/snippet.html');
results.replaceWith(mysnippet);
```

Same result as the previous script, but a different notation:

```
results.replaceWith(loadhtml('snippets/snippet.html'));
```

The following script loads a snippet into a variable and finds/replaces text in the variable before inserting the content into the page. The second `find` command also adds formatting to the replacing text.

```
var mysnippet = loadhtml
('file:///C:/Users/PParker/Documents/Example.html');
mysnippet.find('@var1@').text('OL Connect 1');
mysnippet.find('@var2@').html('<i>OL Connect 2</i>').css('text-
decoration','underline');
results.replaceWith(mysnippet);
```

This last script loads a remote snippet into a variable and retrieves an element from the snippet using query().

```
var mysnippet = loadhtml('http://www.somewebsite.com/text-root-
wrapped.html');
var subject = query("#subject", mysnippet).text();
results.append("<p style='font-weight: bold;'>" + subject +
"</p>");
```

loadhtml(location, selector)

Retrieves specific content from the specified HTML file.

location

String containing a path that can be absolute or relative to the section/context.
Use `snippets/<snippet-name>` to retrieve the content from an HTML file residing in the Snippets folder on the Resources panel.
In order to retrieve files from outside the template the `file` protocol is supported as well as the `http/https` protocols.
The complete syntax of a fully qualified URL with the "file" protocol is: file://<host>/<path>. If the host is "localhost", it can be omitted, resulting in file:///<path>, for example: file:///c:/somefolder/somecontent.html.
When using the http/https protocol, remember that only absolute paths are supported inside remote snippets (see "Remote snippets" on page 286).

Tip

To quickly get the location of an HTML snippet in the template resources, right-click the

snippet in the Resources pane and select **Copy Resource Location**.
The path to a remote HTML snippet can be copied from the snippet's properties: right-click the snippet in the Resources pane and select **Properties**.

selector

String. The supplied selector should conform to CSS selector syntax and allows you to retrieve only the content of matching elements.
If the selected element is a Dynamic Table, the retrieved HTML will contain the expanded table.

Examples

This script loads a **specific element** from a snippet and uses that to replace the `results` (the HTML element or set of HTML elements matched by the selector of the script; see "results" on page 972).

```
var mysnippet = loadhtml('snippets/snippet-
selectors.html','#item3');
results.replaceWith(mysnippet);
```

This script loads the **children** of the selected element.

```
var snippet = loadhtml
('file:///C:/Users/PParker/Documents/Example.html','foobar').childr
en();
results.replaceWith(snippet);
```

The next script loads a **remote** snippet (see "Remote snippets" on page 286), looks for an H1 heading and uses that text.

```
var post = loadhtml('snippets/post.rhtml');
var h1 = query('h1', post).text();
results.text(h1);
```

Another example is given in the following how-to: Using a selector to load part of a snippet.

loadjson()

Creates a JSON object based on the text retrieved from the supplied location. The function lets you retrieve content from a JSON enabled server using a standard HTTP request. Popular

content management systems like WordPress (requires JSON API plugin) and Drupal provide a JSON service/API to retrieve content.

Note

- The specified JSON file is expected to be UTF-8, UTF-16 or UTF-32 encoded. A byte order mark specified as the first character will be stripped. Saving any changes will change the encoding to UTF-8.
- Loadjson() is cached per batch run (based on the URL) in print/email.
- This online JSON viewer is handy to debug JSON data: http://jsonviewer.stack.hu

Tip

External content is not loaded while editing a script. To test a script that loads external content, you can do a preflight; see "Doing a Preflight" on page 405.

loadjson(location)

Loads json data from the specified location.

location

String; the supplied location should be either a URL or a relative file path. The `file` protocol is supported as well as the `http/https` protocols. The complete syntax of a fully qualified URL with the "file" protocol is: file:///<host>/<path>. If the host is "localhost", it can be omitted, resulting in file:///<path>, for example: file:///c:/somefolder/somejson.json.

Examples

This sample script retrieves JSON data from a snippet.

```
var localJSON = loadjson('snippets/jsonsnippet.json');
if(localJSON.post){
        results.html("<h3>" + localJSON.post.title + "</h3><p>" +
localJSON.post.modified + "</p>");
}
```

This script retrieves a post from a WordPress site.

```
var wpPost = loadjson('http://192.168.101.58/2013/06/leave-the-
third-dimension-behind-and-focus-on-real-printing-
innovation/?json=1');
if(wpPost.post){
      results.html("<h1>" + wpPost.post.title + "</h1>"
                + wpPost.post.content);
}
```

This script retrieves multiple posts from a WordPress site.

```
var numPosts = 3;
var wpPost = '';
var wpRecentPosts = loadjson('http://192.168.101.58/?json=get_
recent_posts&count=' + numPosts);
if(wpRecentPosts.posts){
        for (var i = 0; i < numPosts ; i++) {
                wpPost += "<p>" + wpRecentPosts.posts[i].title + "</p>";
        }
}
results.after(wpPost)
```

loadtext()

Loads text from the specified text file, for example a JavaScript file (.js) or style sheet (.css). The text file may be located inside the template or hosted on a Content Management System or on another location outside the template.

Tip

To load an HTML fragment, you can use use `loadhtml()` (see Skin/Formats/CrossReferencePrintFormat("loadhtml()Loads HTML content from the specified HTML file. The file may be located inside the template (see Skin/Formats/CrossReferencePrintFormat("Snippets" on page 1)) or hosted on a Content Management System or on another location outside the template.An optional selector allows you to retrieve only the content of matching elements.The specified HTML file is expected to be UTF-8 encoded.Loadhtml() is cached per batch run (based on the URL) in print/email.To load a JavaScript file (.js) or a style sheet (.css) you can use loadtext(). See Skin/Formats/CrossReferencePrintFormat("loadtext()" on page 1).External content is not loaded while editing a script. To test a script that loads external content, you can do a preflight; see Skin/Formats/CrossReferencePrintFormat("Doing a Preflight" on page 1).loadhtml(location)Loads all HTML from the specified HTML file.locationString

containing a path that can be absolute or relative to the section/context. Use snippets/<snippet-name> to retrieve the content from an HTML file residing in the Snippets folder on the Resources panel. In order to retrieve files from outside the template the file protocol is supported as well as the http/https protocols. The complete syntax of a fully qualified URL with the "file" protocol is: file://<host>/<path>. If the host is "localhost", it can be omitted, resulting in file:///<path>, for example: file:///c:/somefolder/somecontent.html.When using the http/https protocol, remember that only absolute paths are supported inside remote snippets (see Skin/Formats/CrossReferencePrintFormat("Remote snippets" on page 1)).To quickly get the location of an HTML snippet in the template resources, right-click the snippet in the Resources pane and select Copy Resource Location.The path to a remote HTML snippet can be copied from the snippet's properties: right-click the snippet in the Resources pane and select Properties.ExamplesThis script loads a local HTML snippet (from the Resources panel) directly into the matched elementsresults.loadhtml ("snippets/snippet.html");The following script loads a local HTML snippet (Resources panel) into a variable. The replaceWith() command is used to replace the element(s) matched by the script's selector with the contents of the snippet.var mysnippet = loadhtml ('snippets/snippet.html'); results.replaceWith(mysnippet);Same result as the previous script, but a different notation:results.replaceWith(loadhtml('snippets/snippet.html'));The following script loads a snippet into a variable and finds/replaces text in the variable before inserting the content into the page. The second find command also adds formatting to the replacing text.var mysnippet = loadhtml ('file:///C:/Users/PParker/Documents/Example.html'); mysnippet.find('@var1@').text('OL Connect 1'); mysnippet.find('@var2@').html('OL Connect 2').css('text-decoration','underline'); results.replaceWith(mysnippet); This last script loads a remote snippet into a variable and retrieves an element from the snippet using query().var mysnippet = loadhtml('http://www.somewebsite.com/text-root-wrapped.html'); var subject = query("#subject", mysnippet).text(); results.append(" " + subject + " ");loadhtml(location, selector)Retrieves specific content from the specified HTML file.locationString containing a path that can be absolute or relative to the section/context. Use snippets/<snippet-name> to retrieve the content from an HTML file residing in the Snippets folder on the Resources panel. In order to retrieve files from outside the template the file protocol is supported as well as the http/https protocols. The complete syntax of a fully qualified URL with the "file" protocol is: file://<host>/<path>. If the host is "localhost", it can be omitted, resulting in file:///<path>, for example: file:///c:/somefolder/somecontent.html.When using the http/https protocol, remember that only absolute paths are supported inside remote snippets (see

Skin/Formats/CrossReferencePrintFormat("Remote snippets" on page 1)).To quickly get the location of an HTML snippet in the template resources, right-click the snippet in the Resources pane and select Copy Resource Location.The path to a remote HTML snippet can be copied from the snippet's properties: right-click the snippet in the Resources pane and select Properties.selectorString. The supplied selector should conform to CSS selector syntax and allows you to retrieve only the content of matching elements.If the selected element is a Dynamic Table, the retrieved HTML will contain the expanded table.ExamplesThis script loads a specific element from a snippet and uses that to replace the results (the HTML element or set of HTML elements matched by the selector of the script; see Skin/Formats/CrossReferencePrintFormat("results" on page 1)).var mysnippet = loadhtml('snippets/snippet-selectors.html','#item3'); results.replaceWith (mysnippet);This script loads the children of the selected element.var snippet = loadhtml ('file:///C:/Users/PParker/Documents/Example.html','foobar').children(); results.replaceWith(snippet);The next script loads a remote snippet (see Skin/Formats/CrossReferencePrintFormat("Remote snippets" on page 1)), looks for an H1 heading and uses that text.var post = loadhtml('snippets/post.rhtml');var h1 = query ('h1', post).text();results.text(h1);Another example is given in the following how-to: Using a selector to load part of a snippet." on page 1)); for JSON, use `loadjson()` ("loadjson()" on page 827).

Tip

External content is not loaded while editing a script. To test a script that loads external content, you can do a preflight; see "Doing a Preflight" on page 405.

loadtext(location)

Returns the content of a text file. The file extension doesn't have to be .txt. It may also be a JavaScript file (.js) or a style sheet (.css), for instance.

location

String containing a path that can either be a URL or a path that is relative to the section/context.
In order to retrieve files from outside the template the `file` protocol is supported as well as the `http/https` protocols.
The complete syntax of a fully qualified URL with the "file" protocol is: file://<host>/<path>. If

the host is "localhost", it can be omitted, resulting in file:///<path>, for example: file:///c:/somefolder/somecontent.js.

Examples

This script loads a JavaScript file (from the Resources pane) directly into a <script> element.

```
var js = loadtext("js/my-script.js");
results.append(query("<script>").text(js));
```

The following script loads a style sheet (from the Resources pane) into the <style> element.

```
var css = loadtext("css/my-styles.css");
results.append(query("<style type='text/css'>").text(css));
```

logger

This is a global `ScriptLogger` object that allows logging messages such as error, warning or informational messages. The messages will appear in the **Messages** pane (see "Preflight Results and Messages" on page 581 and "Designer User Interface" on page 476).

Methods

These are the methods of the logger object.

Method	Parameters	Description
error()	message: string	Logs an error message
info()	message: string	Logs an informational message
warn()	message: string	Logs a warning message

merge

In Control Scripts, the root level instance of the object `merge` is the entry point from where you can query and change the way contexts are merged. It gives access to the template with all its contexts and sections.
For more information about Control Scripts, see "Control Scripts" on page 429 and "Control Script API" on page 931.
Some of the objects are also useful in Post Pagination Scripts; see "Post Pagination Scripts"

on page 445 and "Post Pagination Script API" on page 967.
For sample scripts, follow the links to the respective objects.

Field	Type	Description
channel	"Channel" on page 964	The final output channel: EMAIL, PRINT or WEB. The channel doesn't change when the output consists of different contexts. When generating an email, for example, the channel is EMAIL, even when merging the Print context to attach it to the email.
"context" on page 969	Context	The context rendered by this merge run. If for one record, different contexts need to be output (for example, when the Print context is attached to an email) a record is merged multiple times: once per context. Per merge run, `merge.context` shows with which context the record is merged.
pagination	"Pagination" on page 991	Contains the total page count and sheet count of all sections in the Print context after pagination.
"section" on page 978	Section	In Standard Scripts, this object defines the section that is being merged. **Note!** In Control Scripts, `merge.section` is not available. To retrieve any section in a Control Script, use: `merge.template.contexts.ContextType.Section['Section name'];` (for example: `merge.template.contexts.PRINT.sections["Section EN"]`). In Post Pagination Scripts, only Print sections are available.
"template" on page 960	Template	This object contains the template and all of its contexts. It can be used to find out which contexts are

Field	Type	Description
		available in the template, using `merge.template.contexts` (see "context" on page 969) and to manipulate the sections in those contexts (see "section" on page 978).

template

The `template` object represents the template with all its contexts and sections. It is used frequently in Control Scripts (see "Control Scripts" on page 429 and "Control Script API" on page 931) but it can also be used in Standard Scripts.

It is retrieved via the merge object: `merge.template` (see "merge" on page 989).

Which contexts are available in the template can be queried using `merge.template.contexts`. To get access to a specific context, you have to specify the ContextType (see "ContextType" on page 965).

Field	Type	Description
contexts	Array	Array of contexts (see "context" on page 969) available in the template. The contexts contain the sections (see "section" on page 978).
images	Array	The list of image resources (see "ImageResource" on page 965) included in the template.
masterpages	Array	Array of Master Pages (see "masterpage" on page 934) available in the template.
"media" on page 937	Array	Media available to this template (see "Media" on page 159). For each of them you can specify, enable and position the stationery's front and back.

Field	Type	Description
"properties" on page 961	Properties	This object contains all default properties of the template as well as any custom properties. (On the menu, select **File > Properties** to view and complement the file properties. See "File Properties dialog" on page 496).

Example

The following Control Script retrieves two Print sections. Then, depending on a value in the current record, it enables one section or the other, so that only one of the two sections appears in the output.

```
var printSections = merge.template.contexts.PRINT.sections;
printSections['Section EN'].enabled = false;
printSections['Section FR'].enabled = false;

if(record.fields.Language === 'FR'){
	printSections['Section FR'].enabled = true;
} else {
	printSections['Section EN'].enabled = true;
}
```

properties

The `properties` object inside the `template` object (see "template" on page 960) contains all default properties of the template file as well as any custom properties.

To view and complement the file properties, select **File > Properties** on the menu. See "File Properties dialog" on page 496.

Following are the default properties.

Field	Type	Description
application	String	Application version when the document was last saved

Field	Type	Description
author	String	Original author of the document
company	String	company name
created	Date	Date and time on which the document was created
customProperties	List	A list of defined custom properties
description	String	Description of the document
file	String	File name of the document
keywords	String	Semicolon-separated list of keywords
modified	Date	Date and time on which the document was last saved

Example

This script stores a default property (author) and a custom property in variables.

```
var author = merge.template.properties.author;
var myProperty = merge.template.customProperties.myPropertyName;
```

ImageResource

The `ImageResource` object represents an image resource in a template.

In script, an `ImageResource` is retrieved via the `images` Array which contains a list of all image resources that are included in the current template (see "template" on page 960).
For example: `var myImage = merge.template.images[0];`.

Functions and fields

Field	Type	Description
modified	Number	Timestamp of the last modification of the image resource, for example: 1566660504000.
name	String	The base name of the image resource, for example: "cat.jpg".
path	String	The path of the image resource, relative to the template, for example: "images/cat.jpg".

Example

The following script adds an image called 'dog.jpg' to the content of the template if the image can be found in the template's resources:

```
const image = merge.template.images.find(image => image.name == 'dog.jpg');
if (image) {
 results.after(`<img src=${image.path}>`);
}
```

masterpage

The `masterpage` object is used to set a Master Page's header and footer or to replace its entire HTML body. It can **only** be accessed in Control Scripts (see "Control Scripts" on page 429 and "Control Script API" on page 931). Trying to access it in another type of script will result in an error.
Note that Master Pages are only used in a Print context (see "Master Pages" on page 156).

In script, Master Pages are retrieved via the masterpages Array which contains all Master Pages in the current template (see "template" on page 960).
For example: `var myMasterpage = merge.template.masterpages["Master page 1"];`.

Functions and fields

Field	Type	Description
"html()" on page 958		Gets or sets the HTML of the body element.
"margins" on page 987	Margins	Allows to set the header and footer of a Master Page, using a Measurement string, for example: "2in" (this sets a margin to two inches).

html()

In a Control Script, the `html()` function of a section or Master Page can be used to get the initial contents of its <body>, and modify them. This makes it possible, for example, to populate a section or Master Page with elements retrieved from a Content Management system, before Standard Scripts run.

html()

Gets the initial contents of the <body> of a section or Master Page.

html(value)

Replaces the contents of the <body> of a section or Master Page with the supplied value. This function is only available in Control Scripts. See also: "section" on page 978 and "masterpage" on page 934.

value

A String that may contain HTML tags.

Examples

The following script uses `html()` to retrieve the contents of a section and add a paragraph to it.

```
var foo = merge.context.sections["Section 1"].html();
foo += "<p>hello world</p>";
merge.context.sections["Section 1"].html( foo );
```

The following script loads a snippet based on the value of a field, and then replaces the content of a Print section with the snippet using `html()`.

```
var mySection = merge.context.sections["Section 1"];
var promoTxt = loadhtml('snippets/promo-' + record.fields['City'] +
'.html');
mySection.html(promoTxt);
```

margins

The `margins` object is used to set a Print section or Master Page's margins in a Control Script (see "Control Scripts" on page 429 and "Control Script API" on page 931).
Note that Master Pages are only used in a Print context (see "Master Pages" on page 156).

The `margins` object is retrieved via the `section` object (see "section" on page 978) or via the `masterpages` array in a template (see "masterpage" on page 934), respectively.

Fields

Field	Type	Description
bottom **left** **right** **top**	String	These fields allow to set the bottom, left, right and top of a Print section, using a Measurement string, for example: "2in" (this sets a margin to two inches).
footer **header**	String	These fields allow to set the header and footer of a Master Page, using a Measurement string.

Examples

Setting the margins of a Print section

```
var sectionA = merge.template.contexts.PRINT.sections["Section A"];
sectionA.margins.top = "2in";
sectionA.margins.left = "2in";
sectionA.margins.right = "2in";
sectionA.margins.bottom = "2in";
```

Setting the header and footer of a Master Page

```
var masterA = merge.template.masterpages["Master page A"];
masterA.margins.header = "2in";
masterA.margins.footer = "2in";
```

media

The media object can be used to specify, enable and position a stationery's front and back in a Control Script (see "Control Scripts" on page 429 and "Control Script API" on page 931). Note that Media are only used in Print sections (see "Media" on page 159).

The available media are listed in, and retrieved via, the template object (see "template" on page 960), for example: `var myMedia = merge.template.media.My_Media`.

Fields

Field	Type	Description
height	String	The height of the Media using a Measurement string. For example, "11in" is 11 inches. In a Control Script this property can be used to change the height of a Media. In other scripts it is only possible to get the height of the Media (read-only).
stationery	Stationery	The Stationery's object's `front` and `back` fields are used to set the front and the back of a Media; see "front, back" on page 937.
width	String	The width of the Media using a Measurement string. For example, "8.5in" is 8.5 inches. In a Control Script this property can be used to change the width of a Media. In other scripts it is only possible to get the width of the Media (read-only).

front, back

The `front` and `back` fields of the Stationery object are used to set the front and the back of a Media (see "media" on page 937).
Both `front` and `back` have the following fields.

Field	Type	Description
enabled	boolean	When enabled, the stationery will be included in the output (Print sections only).
url	String	Specifies the image to use as virtual stationery. For a file named My_Media.pdf, stored inside the template resources, the url would be `images/My_Media.pdf`. The complete syntax for an external file is: **file://<host>/<path>**. If the host is "localhost", it can be omitted, resulting in **file:///<path>**, for example: `file:///c:/users/Adminstrator/Pictures/My_Media.jpg`.
page	Number	PDF and TIFF files may count more than one page. Specify the page number to use.
position	"MediaPosition" on page 966	The `position` is used to place the stationery on the page (absolute, centered, fit to media).
top	Measurement	The vertical offset from the top of the page, used to position the stationery (only when absolute positioning is selected). This value can be negative.
left	Measurement	The horizontal offset from the left of the page, used to position the stationery (only when absolute positioning is selected). This value can be negative.

Number functions

> **Note**
>
> The locale also influences the output of some Number functions; see "Locale" on page 336.

currency(value)

Formats a number as an amount of money. Which currency symbol and which thousands separator are used depends on the Locale; see "Locale" on page 336.

value

A number. Note that all data fields contain strings. Strings need to be converted to a number before passing them to this function.

currency(value, pattern)

Formats a number as an amount of money using a custom pattern. Which currency symbol and which thousands separator are used depends on the Locale; see "Locale" on page 336. For available patterns, see "Number patterns" on the facing page.

value

A number. Note that all data fields contain strings. Strings need to be converted to a number before passing them to this function.

pattern

A custom pattern that may consist of symbols; see "Number patterns" on the facing page. Note that the repetition of pattern letters plays a part in determining the exact presentation.

currencyNoSymbol(value)

Formats a number as a currency whilst omitting the currency symbol.

value

A number. Note that all data fields contain strings. Strings need to be converted to a number before passing them to this function.

grouped(value)

Formats a number using a thousands separator. Which separator is used depends on the Locale, see "Locale" on page 336.

value

A number. Note that all data fields contain strings. Strings need to be converted to a number before passing them to this function.

Number patterns

Numbers, used in a template, can be displayed using a custom pattern.

In the Script Editor, the pattern can be passed to a function of the `formatter`; see "formatter" on page 817. The custom pattern may consist of pattern characters (see below), a prefix and a suffix.

All data fields contain strings.

Note that strings need to be converted to a number before they can be formatted this way.

The repetition of pattern letters determines the exact presentation. For example, the pattern "00000" limits the number to 5 digits and adds leading zeros to any numbers that are not 5 digits long.

Pattern characters

Symbol	Location	Localized?	Meaning
0	Number	Text	Digit
#	Number	Year	Digit, zero shows as absent
.	Number	Year	Decimal separator or monetary decimal separator
-	Number	Month	Minus sign

,	Number	Number	Grouping separator
E	Number	Number	Separates mantissa and exponent in scientific notation. Need not be quoted in prefix or suffix.
;	Subpattern boundary	Number	Separates positive and negative subpatterns
%	Prefix or suffix	Number	Multiply by 100 and show as percentage
\u2030	Prefix or suffix	Number	Multiply by 1000 and show as per mille value
¤ (\u00A4)	Prefix or suffix	Number	Currency sign, replaced by currency symbol. If doubled, replaced by international currency symbol. If present in a pattern, the monetary decimal separator is used instead of the decimal separator.
'	Prefix or suffix	Text	Used to quote special characters in a prefix or suffix, for example, "'#'#" formats 123 to "#123". To create a single quote itself, use two in a row: "# o''clock".

Source: http://docs.oracle.com/javase/7/docs/api/java/text/DecimalFormat.html.

query()

This function creates a new result set, containing the HTML elements in the current section that match the supplied CSS selector. The context (optional) allows you to restrict the search to descendants of one or more context elements.

The new result set is of the type `QueryResults`, just like the `results` object which is also the result of a (hidden) query. All functions that can be used with the `results` object can also be used with this result set; see "results" on page 972.

Note

The `query()` function can't be used in a Control Script, since Control Scripts don't have access to the DOM.

query(selector)

Creates a new result set containing the HTML elements in the current section that match the supplied CSS selector.

selector

A String containing a CSS selector. See https://www.w3schools.com/cssref/css_selectors.asp for CSS selectors and combinations of CSS selectors.

Examples

Look for an element with a certain ID

This scripts applies a style rule to the queried elements.

```
query("#test1").css("color", "yellow");
```

Matched element	Matched element after script execution
foo	foo

Look for an element in a snippet

The following script loads a snippet. Then it looks up an element in a snippet and sets its text. Finally, it replaces the elements matched by the script's selector by the snippet.

```
var snippet = loadhtml('snippets/mysnippet.html');
query("#foo", snippet).text("bar");
results.replaceWith(snippet);
```

query(selector, context)

Creates a new result set containing the HTML elements that match the supplied CSS selector. The context (optional) allows you to restrict the search to descendants of one or more context elements.

selector

A String containing a CSS selector. See https://www.w3schools.com/cssref/css_selectors.asp for CSS selectors and combinations of CSS selectors.

context

A result set (the result of another query) or an HTML string. If the passed context is not a result set or HTML string it will be coerced to a String and interpreted as HTML.

Examples

This script performs a query in the results of another query.

```
var table = query("table");
var rows = query("tr", table);
var cells = query("td", rows);
```

Since the `results` object also is the result of a query (for elements that match the selector of the script), it can be passed as context. For example in a script with the selector “table”.

```
var rows = query("tr", results);
```

query(html)

Creates a new HTML element on the fly from the provided string of raw HTML, and returns the new element.

html

A String containing a HTML element. Tags that can contain other elements should be paired with a closing tag.

Example

The following script adds a paragraph to the `results` (elements that match the selector of the script).

```
var div = query( '<div>' );
for ( let i = 0; i < 5; i++ ){
        var p = query( '<p>' );
        p.text( 'This is number ' + i );
        div.append( p );
}
results.after( div );
```

Tip

The Dynamic Attachment script uses this function to add an attachment to an Email section; see "Email attachments" on page 187.

record

The record object gives access to the record that is currently being merged with the template.

Properties

Field	Type	Description
<fieldName>	Object	Shortcut to access a field in the fields array. For example, you can use `record.FirstName` to access `record.fields.FirstName`. **Note** If a record has a field and a table of the **same name**, `record.<fieldName>` will return the table.
fields	Array	The field values that belong to this record.
id	Number	The id of this record.
index	Number	The one-based index of this record, or zero if no data is available.
<tableName>	Array of records	Shortcut to access a table in the tables array. For example, you can use `record.Products` to access `record.tables.Products`.
tables	Array	The detail tables that belong to this record.

Accessing fields and tables

You can access a specific **field** value:

- using the field's name: `record.fieldName` (case sensitive)
- via the `fields` array, by name: `record.fields.fieldname` or `record.fields['fieldname']` (case insensitive)

If there is no field with the specified name, `record.fields.fieldname` returns an empty string, whereas `record.fieldname` will return `undefined` (unless there is a table with the same name, in which case the table is returned).

A **detail table** can be accessed using its name: `record.tableName`, or via the `tables` array. Either way the table name should be followed by a numeric index for a record in that detail table.

For example, to access the value of the field "prod_id" in the first record of a detail table called "detail", you can use:
`record.detail[0].fields.prod_id` or
`record.tables["detail"][0].fields.prod_id` or
`record.tables.detail[0].fields.prod_id` or
`record.detail[0].prod_id`.

If there is no table (and no field) with the specified name, `record.tableName` results in `undefined`.

In order to **loop** over records in a detail table you could use a for(... in ...) loop (see "for(... in ...)" on page 816), for example:

```
var records = record.tables.detail;
for (var i in records) {
        var rec = records[i];
        ...
}
```

Alternatively you could use an 'Each matched element' script (see "Setting the scope of a script" on page 399).

Yet another way is to create a standard `for` loop using the table's `length` property:

```
var records = record.tables.detail;
for (var i = 0; i < records.length; i++) {
    var rec = records[i];
    ...
}
```

Examples

record.fields

The following Standard Script evaluates the data field *Country* in the current record. If the value is 'CANADA' it will show the results, otherwise it will hide them. (The results object contains the elements that match the script's selector; see "results" on page 972 and "Writing your own scripts" on page 393.)

```
if (record.fields["Country"] == "CANADA") {
        results.show();
} else {
        results.hide();
}
```

Instead of `record.fields["Country"]` you could write `record.Country` (case sensitive), if there is no detail table with the same name.

In a Control Script, an entire section could be enabled or disabled based on the same condition:

```
if (record.Country == "CANADA") {
        merge.template.contexts.PRINT.sections["Section 1"].enabled =
true;
} else {
        merge.template.contexts.PRINT.sections["Section 1"].enabled =
false;
}
```

(For more information about Control Scripts, see "Control Scripts" on page 429.)

record.tables

The next script looks up a value in the first record in a detail table called "detail" and shows or hides the results depending on that value.

```
if (record.tables.detail[0].fields.prod_id == "10") {
        results.show();
```

```
} else {
        results.hide();
}
```

Note that indexes start counting at 0, so `tables.detail[0]` refers to the first record in the detail table.

results

The `results` object (type: `QueryResults`) is the result of the query for HTML elements that match the selector of the script. The selector of a script can be specified in the Script Editor and is visible in the second column of the Scripts pane, next to the name of the script.

If, for example, a script would have the selector `p.onlyCanada`, the script would apply to all paragraphs that have the class `onlyCanada`. (Classes can be defined in the **Attributes** pane at the right: select the element in the content and type the class(es) in the **Class** field.)
The script could then use the `results` object to hide or show those paragraphs, depending on the value of the data field `Country` in the current record:

```
if (record.fields["Country"] == "CANADA") {
results.show();
} else {
results.hide();
}
```

> **Note**
>
> This object can't be used in Control Scripts, because they don't have a selector.

> **Tip**
>
> The easiest way to access the elements in a result set one by one, is by setting the scope of the script to "Each matched element" and using the this object (see "this" on page 891).

Property

Field	Type	Description
length	Number	Number of elements in this result set. Equivalent to calling size ().

Functions

The functions below can be called by the `results` object and by any other result set that is returned by a query, see "query()" on page 845.

Function	Description
"add()" on page 896	Adds elements to a set of HTML elements.
"addClass()" on page 897	Adds the specified class to each element in a set of HTML elements. Has no effect if the class is already present.
"after()" on page 898	Inserts content after each element in a set of HTML elements..
"append()" on page 900	Inserts content at the end of each element in a set of HTML elements.
"attr()" on page 903	Change the given attribute of the element or set of HTML elements with the given value.
"before()" on page 904	Inserts content before an element or before each element in a set of HTML elements.
"children()" on page 907	Returns the immediate children of an HTML element.
"clone()" on page 957	Returns a new result set containing a copy of each element in a set of HTML elements.

Function	Description
"closest()" on page 908	For each element in a set, this function gets the first parent element that matches a selector, by testing the element itself and traversing up through its ancestors in the DOM tree.
"css()" on page 909	Gets the value of a style property for the first element in set of HTML elements or sets one or more CSS properties for every element in a set of HTML elements.
"empty()" on page 912	Removes the contents (child elements and inner HTML) from one element or a set of elements in the template.
"filter()" on page 872	Returns a subset of the current result set.
"find()" on page 912	Performs a search for a text in the children of each element in a set of HTML elements, and returns a new result set with elements that surround the occurrences.
"get(index)" on page 873	Returns the element (type: QueryResult) found at the supplied index in a set of HTML elements.
"hasClass()" on page 913	Returns `true` if the first element in this result set has the specified class.
"height()" on page 913	Gets or sets the outer height of an element, including padding and borders.
"hide()" on page 914	Hides the HTML element or set of HTML elements.
"html()" on page 915	Replaces the inner HTML of the element or of each element in a set of HTML elements with the supplied value, or returns the HTML of the first element if no value is supplied.
"info()" on page 977	**Post Pagination Scripts only**. Returns pagination information for the first element in this result set.

Function	Description
is(selector)	Returns true if at least one of the elements in a set of HTML elements matches the supplied CSS selector.
"next()" on page 916	Returns the next sibling of each HTML element in the result set.
"overflows()" on page 977	Returns a boolean value indicating whether an HTML element overflows its box boundaries.
"pageref()" on page 919	Returns a **marker** that will be replaced with the element's page number after pagination. This only works for elements in the section that is currently being merged.
"parent()" on page 920	Returns the parents of the elements in a set of HTML elements.
position()	Returns information about the position (see NodePosition) of the first element in this result set, excluding margins. To include margins, call `position(true)`.
"prepend()" on page 921	Inserts content at the beginning of an HTML element or of each element in a set of HTML elements.
"prev()" on page 923	Returns the previous sibling of each HTML element in the result set.
"remove()" on page 925	Removes an HTML element or a set of HTML elements from the document.
"removeAttr()" on page 926	Removes the specified attribute from each element in this result set.
"removeClass()" on page 926	Removes the specified class from an element or from each element in a set of HTML elements. Has no effect if the class is not present.
"replaceWith()"	Replaces an HTML element or a set of HTML elements (with a snippet,

Function	Description
on page 927	for example). Returns the result set.
"show()" on page 928	Shows the HTML element or a set of HTML elements.
size()	Gets the number of elements in this result set. Equivalent to the `length` property.
"tagName()" on page 928	Returns the HTML tag name of the first element in this result set, in uppercase. For an example see: "Creating a Table Of Contents" on page 447.
"text()" on page 929	Replaces the text content of an HTML element or of each element in a set of HTML elements with the supplied value, or returns the text content of the first element if no value is supplied.
"width()" on page 929	Gets or sets the outer width of an element, including padding and borders.

add()

The add() function adds elements to one or more HTML elements, which can be:

- The elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The elements returned by a query in the template (see "query()" on page 845).

add(content)

The add() function adds HTML content to one or more HTML elements and returns the union of this result or result set and other content.

content

A query result. This can be an HTML string or a result set.

Examples

Add one result set to another

This script adds one query result to another and sets the background color to yellow.

```
query("#test1").add(query("#test2")).css("background", "yellow");
```

Note: the way the functions add() and css() are used in this script is called 'chaining'. Chaining is optional; the same could be achieved by storing the results of the queries in a variable:

```
var myResult = query("#test1");
myResult.add(query("#test2");
myResult.css("background", "yellow");
```

Creating an empty result set and adding elements to it

The following script loads snippets in an iteration and adds their elements to an empty result set (using `query()`). Then it replaces a placeholder in the template with the new result.

```
var chapters = query();
for ( var i = 1; i <= 4; i++) {
chapters = chapters.add(loadhtml('snippets/Chapter' + i +
'.html'));
}
results.replaceWith(chapters);
```

Selector	Matched element	Matched element after script execution
#chapters	 {{chapters}}	Chapter 1 Lorem ipsum... Chapter 2 Lorem ipsum... Chapter 3 Lorem ipsum... Chapter 4 Lorem ipsum...

addClass()

Adds the specified class(es) to one ore more HTML elements, which can be:

- The elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The elements returned by a query in the template (see "query()" on page 845).

Adding a class has no effect if the class is already present.

addClass(classname)

Adds the specified class(es) to one or more HTML elements. This has no effect if the class is already present.

classname
String, space separated list of class names.

Examples

This script adds a class name to a paragraph.

```
results.addClass("foo");
```

Selector	Matched element	Matched element after script execution
p	Hello world	Hello world

The following script adds two class names to a paragraph.

```
results.addClass("foo bar");
```

Selector	Matched element	Matched element after script execution
p	Hello world	Hello world

after()

Inserts content after one or more HTML elements, which can be:

- The elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The elements returned by a query in the template (see "query()" on page 845).

See also: "before()" on page 904.

after(content)

Inserts content after one or more HTML elements and creates a new result set.

content

String, HTML string or result set to insert after the matched elements. In case a plain text string is provided, it is automatically wrapped in a element to avoid orphan text nodes to appear in the <body> element.

Examples

This script looks up an element with the ID #salesrep and inserts a paragraph after it.

```
query("#salesrep").after("<p>Lorem ipsum</p>");
```

Matched element	Matched element after script execution
Peter Parker	Peter Parker Lorem ipsum

This script looks up an element with the ID #salesrep, sets its text color to red and inserts a paragraph after it.

```
query("#salesrep").after("<p>Lorem ipsum</p>").css("color","red");
```

Matched element	Matched element after script execution
Peter Parker	Peter Parker Lorem ipsum

Note: the way the functions after() and css() are used in this script is called 'chaining'. Chaining is optional; the same could be achieved by storing the result of the query in a variable:

```
var salesrep = query("#salesrep");
salesrep.after("<p>Lorem ipsum</p>");
salesrep.css("color","red");
```

The following script inserts a paragraph after the elements in the results (the set of HTML elements that match the selector of the script).

```
results.after("<p>Lorem Ipsum</p>");
```

Matched element	Matched element after script execution
Peter Parker	Peter Parker Lorem ipsum

This script looks for the string "Lorem " in the results (the set of HTML elements that match the selector of the script) and inserts the string "ipsum" right after that text. The string is automatically enclosed in a span.

```
results.find("Lorem ").after("ipsum");
```

Matched element	Matched element after script execution
Lorem dolor sit amet, consectetur adipiscing elit.	Lorem ipsum dolor sit amet, consectetur adipiscing elit.

This script looks up an element with the ID #salesrep and inserts a string after it. The string is automatically enclosed in a span.

```
query("#salesrep").after("Lorem Ipsum");
```

Matched element	Matched element after script execution
Peter Parker	Peter Parker Lorem Ipsum

append()

Insert content at the end of one or more HTML elements, which can be:

- The elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The elements returned by a query in the template (see "query()" on page 845).

See also: "prepend()" on page 921.

append(content)

Inserts content as the last element at the end of one or more HTML elements. `Append()` creates a new result set.

content

String, HTML string or result set to insert after the element(s). In case a plain text string is provided, it is automatically wrapped in a element to avoid orphan text nodes to appear in the <body> element.

Examples

This script appends a paragraph to the `results` (the set of HTML elements that match the selector of the script).

```
results.append("<p>Peter Parker</p>");
```

Selector	Matched element	Matched element after script execution
#box		

Selector	Matched element	Matched element after script execution
	Personal information 	Personal information Peter Parker

This script appends a string to the `results` (the HTML elements that match the selector of the script). The string is added to the end of the matched element(s) and wrapped in a Span element.

```
results.append("Peter Parker");
```

Selector	Matched element	Matched element after script execution
.name	 Personal information Name: 	 Personal information Name:Peter Parker

This script's selector is , so the script appends a paragraph to all Div elements in the template.

```
results.append("<p>Peter Parker</p>");
```

Selector	Matched element	Matched element after script execution
div	 Personal information Personal information 	 Personal information Peter Parker Personal information Peter Parker

The following script appends a snippet to a Div element with the ID `box`.

```
var a = loadhtml('snippets/snippet_name.html');
results.append(a);
```

Selector	Matched element	Matched element after script execution
#box	 Personal information 	 Personal information Peter Parker

This script looks for an element with the ID `box` and appends a paragraph to it.

```
query("#box").append("<p>Peter Parker</p>");
```

Matched element	Matched element after script execution
  Personal information 	 Personal information Peter Parker

This script looks for an element with the ID `box`, appends a paragraph to it and colors all text inside the box red.

```
query("#box").append("<p>Peter Parker</p>").css("color","red");
```

Matched element	Matched element after script execution
  Personal information 	 Personal information Peter Parker

Note: the way the functions append() and css() are used in this script is called 'chaining'. Chaining is optional; the same could be achieved by storing the result of the query in a variable:

```
var box = query("#box");
box.append("<p>Peter Parker</p>");
box.css("color","red");
```

attr()

attr(attributeName) : String

Returns the value of the specified attribute of an HTML element which can be:

- The first element in a set of elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The first element in a set of elements returned by a query in the template (see "query()" on page 845).

attributeName

String; the name of the attribute.

Examples

This script - with the selector `img` - stores the source of the first image in a variable.

```
var src = results.attr("src");
```

The following script looks up an image with the ID #image1 and stores its background color in a variable.

```
var imgURL = query("#image1").attr("src");
```

attr(attributeName, value)

Sets the value of the specified attribute of one HTML element or of each element in a result set.

attributeName

String; the name of the attribute.

value

String; value for the attribute.

Examples

This script looks up an image in an element with the ID #calloutbox and sets its alternative text to a value from a data field.

```
var altText = record.fields.FavHobby;
query("#callout img").attr('alt', altText);
```

The following script sets the background color of a specific table cell in an email to red if the value of the field TOTAL has a negative value in the current record.

```
if(record.fields.TOTAL<0) {
        query("#total").attr("bgcolor","red");
}
```

before()

Inserts content before one or more HTML elements, which can be:

- The elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The elements returned by a query in the template (see "query()" on page 845).

See also: "after()" on page 898.

before(content)

Before(content) inserts content before one or more HTML elements and creates a new result set.

content

String, HTML string or result set to insert after the elements. In case a plain text string is provided, it is automatically wrapped in a element to avoid orphan text nodes to appear in the <body> element.

Examples

This script looks for an element with the ID `salesrep`and inserts a paragraph before that element.

```
results.before("<p>Lorem Ipsum</p>");
```

Selector	Matched element	Matched element after script execution
#salesrep	Peter Parker	Lorem ipsum Peter Parker

This script does the same, but it uses the query() function to look up the element.

```
query("#salesrep").before("<p>Lorem ipsum</p>");
```

Matched element	Matched element after script execution
Peter Parker	Lorem ipsum Peter Parker

The following script looks for an element with the ID salesrep, inserts a paragraph before that element and colors that element red.

```
query("#salesrep").before("<p>Lorem ipsum</p>").css("color","red");
```

Matched element	Matched element after script execution
Peter Parker	Lorem ipsum Peter Parker

Note: the way the functions `before()` and `css()` are used in this script is called 'chaining'. Chaining is optional; the same could be achieved by storing the result of the query in a variable:

```
var salesrep = query("#salesrep");
salesrep.before("<p>Lorem ipsum</p>");
salesrep.css("color","red");
```

The following script searches the results for the string "ipsum" and puts "Lorem " before it. "Lorem " is automatically wrapped in a Span element.

```
results.find("ipsum").before("Lorem ");
```

Matched element	Matched element after script execution
ipsum dolor sit amet, consectetur adipiscing elit.	Lorem ipsum dolor sit amet, consectetur adipiscing elit.

The following script looks for an element with the ID salesrep and inserts the text "Lorem Ipsum" before that element. "Lorem Ipsum" is automatically wrapped in a Span element.

```
query("#salesrep").before("Lorem Ipsum");
```

Matched element	Matched element after script execution
ipsum dolor sit amet, consectetur adipiscing elit.	Lorem Ipsum Peter Parker

children()

Returns the immediate children (inner HTML) of one or more HTML elements, which can be:

- The elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The elements returned by a query in the template (see "query()" on page 845).

Examples

This script retrieves the inner HTML of an element selected from a snippet.

```
var snippet = loadhtml('snippets/snippet.html','#foobar').children
();
results.append(snippet);
```

The following script retrieves the inner HTML of the elements and then performs a find/replace.

```
var snippet = loadhtml('snippets/snippet.html','#foobar').children
();
snippet.find('@firstname@').text('foobar');
results.append(snippet);
```

clone()

This function returns a copy of one HTML element or of a set of HTML elements, which can be:

- The elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The elements returned by a query in the template (see "query()" on page 845).

See also: "Dynamically adding sections (cloning)" on page 440.

To duplicate an existing template element, clone it before calling `append()`; see "append()" on page 900.

Examples

This script performs an iteration over the elements in the `results` (the elements that match the selector of the script).

```
var row = query("tbody tr", results).clone();
query("tbody", results).append(row);
```

The following script clones an existing table row to match the number of rows in a detail table. Afterwards it iterates over the rows to populate the fields.

```
// Create the number of rows based on the records in the detail
table
// We start at 1 so the boilerplate row is used too and there is no
need to delete that row
for(var r = 1; r < record.tables['detail'].length; r++) {
results.parent().append(results.clone());
}

// Iterate over the rows and populate them with the data from the
accompanying data row
query("#table_2 > tbody > tr").each(function(i) {
this.find('@ItemNumber@').text( record.tables['detail'][i].fields
["ItemNumber"]);
this.find('@ItemOrdered@').text( record.tables['detail'][i].fields
["ItemOrdered"]);
this.find('@ItemTotal@').text( record.tables['detail'][i].fields
```

```
["ItemTotal"]);
this.find('@ItemDesc@').text( record.tables['detail'][i].fields
["ItemDesc"]);
this.find('@nr@').text(i);
});
```

The following script clones and populates a boilerplate row. Once completed you will need to hide the boilerplate row.

closest()

This function gets the first parent element that matches a selector, by testing the element itself and traversing up through its ancestors in the DOM tree. (In HTML, a parent is an element that contains another element.)

The function can be used for:

- The elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The elements returned by a query in the template (see "query()" on page 845).

To get a child element or all child elements, use `children()` (see "children()" on page 907).

The `closest()` command is based on the `closest()` command found in the jQuery library: https://api.jquery.com/closest/.

To get an element's sibling element, you can use "prev()" on page 923 or "next()" on page 916.

closest(selector)

For one HTML element or for each element in a set, this function gets the first element that matches the selector by testing the element itself and traversing up through its ancestors in the DOM tree.

selector

A String containing an HTML tag (without the angle brackets, <>).

Examples

The following script looks up all table rows in the template that contain an <input> element.

```
query("input").closest("tr");
```

This code gets the closest 'parent' row for each element that matches the selector of the script (collected in the `results` object):

```
results.closest("tr");
```

The rows could be colored red within the same statement:

```
results.closest("tr").css('background-color','red');
```

css()

Gets the value of a style property of one HTML element, or sets one or more CSS properties of one or more HTML elements.

css(styleName) : String

Returns the value of the specified CSS property of an HTML element, which can be:

- The first element in a set of elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The first element in a set of elements returned by a query in the template (see "query()" on page 845).

propertyName

String; the name of the CSS property.

Examples

This script stores the text color of the `results` (the HTML elements that match the selector of the script) in a variable.

```
var textcolor = results.css("color");
```

The following script looks up an element with the ID #calloutbox and stores its background color in a variable.

```
var backgroundcolor = query("#calloutbox").css("background-color");
```

css(styleName, value)

Function to set a CSS property of one HTML element or of each element in a result set.

propertyName
String; the name of the CSS property.

value
String; value for the CSS property or a map of property-value pairs to set.

Examples

This script looks up an element with the ID #calloutbox and sets its text color to red.

```
query("#callout p").css('color' , 'red');
```

The following script does the same, but it only sets the text color to red if in the current record the value of the field 'accounttype' is 'PRO'.

```
if(record.fields.accounttype == "PRO") {
query("#callout p").css("color","red");
}
```

This script sets the text color of the results to a hexadecimal color code.

```
results.css('color' , '#669900');
```

This script loads a snippet into a variable. Then it finds/replaces text in the snippet and applies a css property to the replacing text.

```
var mysnippet = loadhtml('snippets/snippet vars.html');
mysnippet.find('@var@').text('OL Connect').css('text-
decoration','underline');
results.replaceWith(mysnippet);
```

css(properties)

Function to set one or multiple CSS properties of one or more HTML elements, which can be:

- The elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The elements returned by a query in the template (see "query()" on page 845).

properties

Array; map of property-value pairs to set.

Examples

This script colors the text of the `results` (the set of HTML elements that match the selector of the script) red and makes it bold.

```
results.css({'color' : 'red', 'font-weight' : 'bold'});
```

empty()

Removes the contents (child elements and inner HTML) from one or more HTML elements, which can be:

- The elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The elements returned by a query in the template (see "query()" on page 845).

Use "remove()" on page 925 to remove the elements themselves.

Example

This script empties all Span elements found in the template.

```
results.empty();
```

Selector	Paragraph before script execution	Paragraph after script execution
span	<p>Lorem ipsum	<p>Lorem ipsum

Selector	Paragraph before script execution	Paragraph after script execution
	dolor sit amet, consectetuer adipiscing elit.</p>	amet, consectetuer adipiscing elit.</p>

filter()

filter(callback)

Returns a subset of a set. All elements for which the callback function returns `true` will be included in the result.

callback

A function used as a test for each element in the set. Filter() passes the iteration index and the current element to the callback function. In the scope of the callback function, `this` refers to the current element.

Example

The selector of the following script is **li** (list item), so the `results` object contains all list items in the template. The scripts filters the third and sixth line items from the `results`, taking advantage of the index that is passed to the filter function, and colors them red. It uses the modulus operator (%) to select every item with an index value that, when divided by 3, has a remainder of 2. (The index starts counting at zero.)

```
results.filter(function(index) {
        return index % 3 === 2;
}).css( "background-color", "red" );
```

filter(selector)

Returns a subset of a set. All elements matching the selector will be included in the result.

The difference between `results.filter(selector)` and `query(selector, results)` is that query() searches throughout the entire `results` while filter() only takes the top-level elements into account.

selector

A String containing a CSS selector. See https://www.w3schools.com/cssref/css_selectors.asp for CSS selectors and combinations of CSS selectors.

Example

The selector of the following script is **tr** (table row), so the object `results` contains all rows in the template. The scripts filters all even rows from the `results` and colors them red.

```
results.filter(":nth-child(even)").css("background-color", "red");
```

find()

find(textToFind)

Performs a deep search for textToFind in the children of each element, and returns a new result set with elements that surround the occurrences.

textToFind

A String that contains the search text.

Example

The following piece of code loads a snippet, then looks for placeholders using find(), and replaces them with a text.

```
var mysnippet = loadhtml('snippets/snippet.html');
mysnippet.find('@var1@').text('OL Connect 1');
mysnippet.find('@var2@').html('<i>OL Connect 2</i>').css('text-
decoration','underline');
results.replaceWith(mysnippet);
```

get(index)

Returns the element found at the supplied index in a set of HTML elements.

index

Place of the element in the result set. The index starts counting at zero.

Example

This Post Pagination Script copies the page numbers of level 1 and 2 headings in all Print sections to the corresponding entry in a table of contents.

```
var $numbers = query('.number');
merge.context.query("h1, h2").each(function( index ) {
        var pageNo = this.info().pageNo;
        var entry = $numbers.get( index );
        if( entry ) {
                entry.text( pageNo );
        }
});
```

This code is used in a script that inserts a table of contents. For the full script and explanation see "Creating a Table Of Contents" on page 447.

hasClass()

hasClass(classname) : Boolean

Returns true if the HTML element or the first element in a result set has the specified class.

classname

String containing one class name.

Example

This script checks if the first of the `results` (the set of elements matching the selector of the script) has the class 'green'. If so, it colors the text of all the elements in the `results` green.

```
if (results.hasClass('green')) {
       results.css('color', 'green');
}
```

height()

Gets or sets the outer height, including padding and borders, of an HTML element, or of the first element in a result set (see "results" on page 972), i.e. the set of HTML elements that match the selector of the script or of another query in the template (see "query()" on page 845).

height(): Number

Returns the outer height of this element, including padding and borders, excluding margins. To include margins, call `height(true)`.

The returned value is the height in pixels, without measurement unit (e.g. 400).

height(value): void

Sets the outer height of this element, including padding and borders, excluding margins. To include margins, call `height(value, true)`.

value

Number. Give the height in pixels, but without the measurement unit (e.g. 400, not 400px).

Examples

This script adds 20 pixels to the width and height of an element.

```
var h = results.height();
var w = results.width();

h = h + 20;
w = w + 20;

results.height( h );
results.width( w );
```

The following script does the same, but in this case, margins are included.

```
var h = results.height(true);
var w = results.width(true);

h = h + 20;
w = w + 20;

results.height( h, true);
results.width( w, true);
```

hide()

Hides one or more HTML elements, which can be:

- The elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The elements returned by a query in the template (see "query()" on page 845).

This doesn't remove the elements; to make them visible again, use the function "show()" on page 928.

These functions are used by the Conditional Script Wizard, as you can see when you open a Conditional Script and click the **Expand** button; see "Showing content conditionally" on page 358.

Example

This script hides or shows the elements matched by the selector of the script (which are stored in the `results` object), depending on the value of the data field `Country` in the current record.

```
if (record.fields["Country"] == "CANADA") {
results.show();
} else {
results.hide();
}
```

html()

html() : String

Returns the inner HTML of an HTML element, which can be:

- The first element in a set of elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The first element in a set of elements returned by a query in the template (see "query()" on page 845).

html(value)

Replaces the inner HTML of an HTML element or of each element in a result set with the supplied value.

value

A String that may contain HTML tags.

Examples

The following script loads part of a snippet based on the value of a field, and then inserts the content into the document using `html()`.

```
var promoTxt = loadhtml('snippets/promo-en.html', '#' +
record.fields['YOGA']);
results.html(promoTxt);
```

The following script loads a snippet. Then it looks for a placeholder (@var2@) in the text of that snippet and replaces every found placeholder by the text 'OL Connect 1'. It uses html() so the HTML formatting (and) will indeed be interpreted as HTML. Finally, it places the snippet in the template.

```
var mysnippet = loadhtml('snippets/snippet.html');
mysnippet.find('@var1@').html('<i>OL Connect 1</i>');
results.replaceWith(mysnippet);
```

info()

Returns pagination information for one HTML element or for the first element in a result set (see "results" on page 972) of a query across all sections in a Print context (see "query(selector)" on page 971).

The returned information is of the type `PaginationInfo` (see "PaginationInfo" on page 971).

This function can only be used in a Post Pagination Script ; see "Post Pagination Script API" on page 967.

For an example see: "Creating a Table Of Contents" on page 447.

next()

`next()` returns the next sibling of an HTML element, which can be:

- The elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element"

(see "this" on page 891 and "Setting the scope of a script" on page 399).

- The elements returned by a query in the template (see "query()" on page 845).

In HTML, siblings are HTML elements that share the same parent, i.e. HTML elements in the same container element.
Note that a sibling can be a different type of element. For example, if a paragraph, an image and a table follow each other in the same Box, they are siblings.

> **Tip**
>
> To get the **previous** sibling, use previous() (see "prev()" on page 923).

What if there are no siblings?

If you call `prev()` or `next()` on a **collection** of elements (either "results" on page 972 or the result set of "query()" on page 845), the function will return a collection of siblings. If none of the elements has a previous or next sibling, the collection will be empty. Actions performed on an empty collection will simply have no effect.

If you call `prev()` or `next()` on a **single** element (this in an 'Each matched element' script), and the element has no previous or next sibling, the function returns `null`. Performing an operation on a `null` value will result in an error, which means that the rest of the script won't be executed. In this case, it is useful to check the return value, unless it's certain that it will never be `null`.

Example

Assume that there are three consecutive paragraphs in a Box and that the second of those paragraphs has an ID that matches the selector of this script. The paragraph is stored in the results object (see "results" on page 972). The following script changes the font-weight of the next paragraph - the third paragraph in the Box - to bold.

```
results.next().css("font-weight", "bold");
```

If the elements in results don't have siblings, this line of code will have no effect. It also won't result in an error.

In an 'Each matched element' script you would first need to check the return value of this.next():

```
var nextSibling = this.next();
if nextSibling != null { nextSibling.css("font-weight", "bold"); }
```

overflows()

The `overflows()` method returns a boolean value indicating whether an HTML element overflows its box boundaries.

This function could for instance be used in a design where content needs to flow separately from the main text flow, or where a new, full-page, multi-column box should be inserted when the current one overflows.

Note that when called in a Standard script, this method runs *before* the pagination of Print output. After pagination - in the output, and in Preview mode - the element's overflow may have changed, especially when the element is combined with floating elements (i.e. elements that have the CSS style float) and located near a page-break.
To prevent this from happening, it may help to set the element's CSS style to `overflow:hidden`.
The `overflows()` method can also be used in a Post Pagination script.

> **Tip**
>
> You don't need a script to resize text in order to make it fit in a box. The Copy Fit feature automatically scales text to the available space (see "Copy Fit" on page 311).

Examples

```
var $box = query("#mybox");
while ( ! $box.overflows()) {
        //do something
}
```

A slightly shorter version:

```
while ( ! query("#mybox").overflows()) {
        //do something
}
```

The following script selects three boxes that all have the class box (selector: .box). It loops over them, inserting as many products from an array as possible into the box. In fact, it inserts one too many and removes the last one before moving on to the next box.

```
var products =
["Appetizer","Beans","Beef","Lettuce","Sprouts","Coconut","Juice","
Soup","Coriander","Cheese","Pasta","Sugar","Vinegar","Bread"];
var i = 0;
```

```
results.each( function() {
        // Add elements to the box until it overflows.
        while ( ! this.overflows() && i < products.length ) {
                this.append("<p>" + products[i] + "</p>");
                i++;
        }
        // Go one step back unless we processed all items in our array.
        if( i < products.length ) {
                query("p:last-child", this).remove();
                i--;
        }
});
```

pageref()

Returns a **marker** that will be replaced with the element's page number after pagination. This only works for elements in the section that is currently being merged.

Example

Creating a table of contents

The following script creates a table of contents for all level 1 headings (<h1> elements) with the class `title` in one section.

```
var toc = '<ul ID="toc">';
query('h1.title').each(function() {
 toc += '<li>' + this.text() + ' <span class="li_toc">' +
this.pageref() + '</span></li>';
});
toc += '</ul>';
results.after(toc);
```

The first line creates a variable for the table of contents, which will be a list (a <ul> element with the ID `toc`). The start tag of the list is added to the variable.
The next line does a query for all level 1 headings (<h1> elements) with the class `title` in the current section. With `each()` the script loops through them. For each of the headings it adds a line item to the list, with the text (`this.text()`) and the page reference of the respective heading.
After the loop, the end tag of the list is added to the variable.
Finally, the script adds the variable - that now contains the table of contents - after the `results`.

The `results` object contains the elements that match the selector of the script. So, if the script's selector selects the title of the table of contents, the table of contents will be added after that.

The following style rules, added to the style sheet, will align the chapter titles to the left and the page numbers to the right:

```
#toc li {
 text-align:left;
}
#toc span {
 float: right;
}
```

Note that these styles use the list's ID, that was defined in the first line of code. For information about style sheets, see "Styling templates with CSS files" on page 296.

parent()

Returns the parent(s) of one or more HTML elements, which can be:

- The elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The elements returned by a query in the template (see "query()" on page 845).

In HTML, a parent is an element that contains another element.
To get an ancestor that matches a particular selector, use `closest()` (see "closest()" on page 908).

To get an element's sibling element, you can use "prev()" on page 923 or "next()" on page 916.

Example

Assume that there are three paragraphs in a Box and that one of those paragraphs matches the selector of this script. The paragraph is stored in the `results` object (see "results" on page 972). The following script retrieves the Box (which is the parent of the paragraph) using `results.parent()`, and then changes its background color to red.

```
results.parent().css('background-color' , 'red');
```

prepend()

Insert content at the beginning of one or more HTML elements, which can be:

- The elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The elements returned by a query in the template (see "query()" on page 845).

See also: "append()" on page 900.

prepend(content)

Insert content as the first element to each element in the set of HTML elements that match the selector of the script or of another query in the template (see "query()" on page 845). `Append` creates a new result set.

content

HTML string, string or HTML string to insert after the matched elements. In case a plain text string is provided, it is automatically wrapped in a element to avoid orphan text nodes to appear in the <body> element.

Examples

This script inserts a heading as the first element in an element that has the ID `#box`.

```
results.prepend("<h1>Personal information</h1>");
```

Selector	Matched element	Matched element after script execution
#box	 Peter Parker 	 Personal information Peter Parker

This script inserts a heading as the first element in an element that has the class `name`.

```
results.prepend("<b>Name: </b>");
```

Selector	Matched element	Matched element after script execution
.name	 Personal information Peter Parker 	 Personal information Name: Peter Parker

This script inserts content in multiple elements at the same time.

```
results.prepend("<h1>Personal information</h1>");
```

Selector	Matched element	Matched element after script execution
div	 Peter Parker Peter Parker 	 Personal information Peter Parker Personal information Peter Parker

This script prepends a snippet that contains the text "<h1>Personal information</h1>".

```
var a = loadhtml('snippets/snippet.html');
results.prepend(a);
```

Selector	Matched element	Matched element after script execution
div	 Peter Parker 	 Personal information Peter Parker

This script uses the function query() to find a box. Then it inserts a heading as the first element in that box.

```
query("#box").prepend("<h1>Personal information</h1>");
```

Matched element	Matched element after script execution
  Peter Parker 	 Personal information Peter Parker

This script uses the function query() to find a box, prepends a heading and sets the text color of the entire box to red.

```
query("#box").prepend("<h1>Personal information</h1>").css
("color","red");
```

Matched element	Matched element after script execution
  Peter Parker 	 Personal information Peter Parker

Note: the way the functions prepend() and css() are used in this script is called 'chaining'. Chaining is optional; the same could be achieved by storing the result of the query in a variable:

```
var box = query("#box");
box.prepend("<p>Peter Parker</p>");
box.css("color","red");
```

prev()

`prev()` returns the previous sibling of an HTML element, which can be:

- The elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The elements returned by a query in the template (see "query()" on page 845).

In HTML, siblings are HTML elements that share the same parent, i.e. HTML elements in the same container element.
Note that a sibling can be a different type of element. For example, if a paragraph, an image and a table follow each other in the same Box, they are siblings.

> **Tip**
>
> To get the **next** sibling, use `next()` (see "next()" on page 916).

What if there are no siblings?

If you call `prev()` or `next()` on a **collection** of elements (either "results" on page 972 or the result set of "query()" on page 845), the function will return a collection of siblings. If none of the elements has a previous or next sibling, the collection will be empty. Actions performed on an empty collection will simply have no effect.

If you call `prev()` or `next()` on a **single** element (this in an 'Each matched element' script), and the element has no previous or next sibling, the function returns `null`. Performing an operation on a `null` value will result in an error, which means that the rest of the script won't be executed. In this case, it is useful to check the return value, unless it's certain that it will never be `null`.

Example

Assume that there are three consecutive paragraphs in a Box and that the second of those paragraphs has an ID that matches the selector of this script. The paragraph is stored in the results object (see "results" on page 972). The following script changes the font-weight of the previous paragraph - the first paragraph in the Box - to bold.

```
results.prev().css("font-weight", "bold");
```

If the elements in results don't have siblings, this line of code will have no effect. It also won't result in an error.

In an 'Each matched element' script you would first need to check the return value of `this.prev()`:

```
var prevSibling = this.prev();
if prevSibling != null { prevSibling.css("font-weight", "bold"); }
```

remove()

Removes the current element or each element in a set from the DOM.

This function returns a new result set containing each removed element. These can be changed and inserted in the document. This could be beneficial in terms of performance, as manipulating elements inside the DOM is relatively time consuming.

Examples

This script removes all Span elements found in the template.

```
results.remove();
```

Selector	Paragraph before script execution	Paragraph after script execution
span	Lorem ipsum dolor sit amet, consectetuer adipiscing elit.	Lorem ipsum amet, consectetuer adipiscing elit.

The selector of the following sample script is `tbody`. Before this script runs, the table body consists of a single placeholder row with three cells. After running the script, it contains thirty rows. To improve performance, most of the DOM manipulation takes place on detached elements.

```
// Detach the placeholder row from the DOM
var row = query("tr", results).remove();
```

```
// Modify the cells of this row
var cells = row.children();
cells[0].html("some text").css("background-color", "yellow");
cells[1].html("some text").css("font-weight", "bold");
cells[2].html("some text");

// Create a number of copies
var rows = row.clone();
for (var i = 0; i < 30; i++) {
       rows = rows.add(row.clone());
}

// Attach all copies to the DOM as children of tbody
results.append(rows);
```

removeAttr()

Removes the specified HTML attribute from an element or from each element in a set of elements. To add or change an attribute, use `attr()` (see "attr()" on page 903).

removeAttr(attributeName)

attributeName

String; the name of the attribute.

Examples

This script looks up an email field in a form (which is an <input> with the ID #email1) and removes its `readonly` attribute.

```
query("#email1").removeAttr('readonly');
```

removeClass()

Removes the specified class from the current HTML element or from each element in a result set. This has no effect if the class is not present.

removeClass(classname)

classname

String, space separated list of class names.

Examples

This script removes the class name "foo" from all elements in the results that have this class.

```
results.removeClass("foo");
```

Selector	Matched element	Matched element after script execution
p	Hello world	Hello world

replaceWith()

Replaces one HTML element or each element in a set of HTML elements.

replaceWith(content)

Replaces one HTML element or each element in a set of HTML elements. Returns the result set.

content

A query result. This can be an HTML string or a result set.

Examples

Replace elements with a snippet

The following script loads a snippet and then replaces the elements matched by the script's selector with the snippet.

```
var snippet = loadhtml('snippets/mysnippet.html');
results.replaceWith(snippet);
```

Replace elements with a set of snippets

The following script loads snippets and adds their elements to a new, empty result set (using `query()`). Then it replaces a placeholder in the template with the set of snippets.

```
var chapters = query();
for ( var i = 1; i <= 4; i++) {
chapters = chapters.add(loadhtml('snippets/Chapter' + i +
'.html'));
```

```
}
results.replaceWith(chapters);
```

show()

Shows one or more HTML elements, which can be:

- The elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The elements returned by a query in the template (see "query()" on page 845).

To hide elements (again), use the function "hide()" on page 914.

These functions are used by the Conditional Script Wizard, as you can see when you open a Conditional Script and click the **Expand** button; see "Showing content conditionally" on page 358.

Example

This script hides or shows the elements matched by the selector of the script (which are stored in the `results` object), depending on the value of the data field `Country` in the current record.

```
if (record.fields["Country"] == "CANADA") {
results.show();
} else {
results.hide();
}
```

tagName()

Returns the HTML tag name of one HTML element in uppercase (e.g. H1, P, ARTICLE).

Example

This Post Pagination Script finds level 1 and 2 headings in all Print sections and stores their page number, text and HTML tag name.

```
merge.context.query("h1, h2").each(function() {
        var pageNo = this.info().pageNo;
        var text = this.text();
        var level = this.tagName();
});
```

This code is used in a script that inserts a table of contents. For the full script and explanation see "Creating a Table Of Contents" on page 447.

text()

text() : String

Returns the text content of an HTML element or of the first element in a result set (see "results" on page 972).

Example

This script loads a snippet into a variable and retrieves an element from the snippet using `query()` and `text()`.

```
var mysnippet = loadhtml('snippets/text-root-wrapped.html');
var subject = query("#subject", mysnippet).text();
results.append("<p style='font-weight: bold;'>" + subject +
"</p>");
```

text(value)

Replaces the text content of one HTML element or of each element in a result set with the supplied value.

Example

This script loads a snippet, then looks for placeholders using find(), and replaces them using text(value).

```
var mysnippet = loadhtml('snippets/snippet.html');
mysnippet.find('@var1@').text('OL Connect 1');
mysnippet.find('@var2@').html('<i>OL Connect 2</i>').css('text-
decoration','underline');
results.replaceWith(mysnippet);
```

width()

Gets or sets the outer width, including padding and borders, of an HTML element or of the first element in result set (see "results" on page 972), i.e. the set of HTML elements that match the selector of the script or of another query in the template (see "query()" on page 845).

width(): Number

Returns the outer width of this element, including padding and borders, excluding margins. To include margins, call `width(true)`.

The returned value is the width in pixels, without measurement unit (e.g. 400).

width(value): void

Sets the outer width of this element, including padding and borders, excluding margins. To include margins, call `width(value, true)`.

value

Number. Give the width in pixels, but without the measurement unit (e.g. 400, not 400px).

Examples

This script adds 20 pixels to the width and height of an element.

```
var h = results.height();
var w = results.width();

h = h + 20;
w = w + 20;

results.height( h );
results.width( w );
```

The following script does the same, but in this case, margins are included.

```
var h = results.height(true);
var w = results.width(true);

h = h + 20;
w = w + 20;

results.height( h, true);
results.width( w, true);
```

this

A Standard script or Post Pagination script that has its scope set to "Each matched element" (see "Setting the scope of a script" on page 399) will be called in a loop over the elements that

match the selector of the script - the `results` (see "results" on page 972). In such a script, `this` is an object of the type QueryResult. It represents the current element in the loop.

> **Note**
>
> The scope of Control Scripts can't be set, because they don't have a selector.

Example

If the selector of a script is `p.onlyCanada`, and its scope is set to "Each matched element", the script will be repeated for all paragraphs that have the `onlyCanada` class; it can access only one paragraph at a time. The script could use `this` to manipulate each paragraph, e.g. hide or show it depending on the value of a data field `Country` in the current record:

```
if (record.fields["Country"] == "CANADA") {
this.show();
} else {
this.hide();
}
```

> **Note**
>
> In a script that has its scope set to "Each matched element", the `results` object is still accessible, and can be used in a non-iterative way (for example, you could use its `length` property to determine the total number of matched elements) but iterating over `results` will produce a warning.

Properties

Field	Type	Description
id	Number	Index in the result set, or zero if this element is not part of a result set.
record	Record	Represents the current detail record in a (possibly nested) detail table associated with repeating rows in a Dynamic Table. This property is only available when

Field	Type	Description
		`this` is a row or child element of a row in a Dynamic Table. (See Dynamic Table.) If `this` (the current QueryResult instance) cannot be matched to a detail record, `this.record` is null.
runningTotal	Number	Can be used to update a running total. It is initialized to zero when a loop over `results` starts.

Functions

The functions below can be called by the `results` object and by any other result set that is returned by a query, see "query()" on page 845.

Function	Description
"add()" on page 896	Adds elements to the current HTML element.
"addClass()" on page 897	Adds the specified class to the current HTML element. Has no effect if the class is already present.
"after()" on page 898	Inserts content after the current HTML element.
"append()" on page 900	Inserts content at the end of the current HTML element.
"attr()" on page 903	Changes the given attribute of the current HTML element with the given value.
"before()" on page 904	Inserts content before the current HTML element.
"children()" on page 907	Returns the immediate children of the current HTML element.

Function	Description
"clone()" on page 957	Returns a copy of the current HTML element.
"closest()" on page 908	This function gets the first parent element that matches a selector, by testing the element itself and traversing up through its ancestors in the DOM tree.
"css()" on page 909	Gets the value of a style property for the HTML element or sets one or more CSS properties for the HTML element.
"empty()" on page 912	Removes the contents (child elements and inner HTML) from the current element.
"find()" on page 912	Performs a search for a text in the children of the current HTML element and returns a new result set with elements that surround the occurrences.
"hasClass()" on page 913	Returns `true` if the current HTML element has the specified class.
"height()" on page 913	Gets or sets the outer height of an element, including padding and borders.
"hide()" on page 914	Hides the HTML element.
"html()" on page 915	Replaces the inner HTML of the element with the supplied value, or returns the HTML of the current element if no value is supplied.
"info()" on page 977	**Post Pagination Scripts only**. Returns pagination information for the current HTML element.
is(selector)	Returns true if the current HTML element matches the supplied CSS selector.
"next()" on	Returns the next sibling of the current HTML element.

Function	Description
page 916	
"overflows()" on page 977	Returns a boolean value indicating whether the current HTML element overflows its box boundaries.
"pageref()" on page 919	Returns a **marker** that will be replaced with the element's page number after pagination. This only works for elements in the section that is currently being merged.
"parent()" on page 920	Returns the parent of the HTML element.
"prepend()" on page 921	Inserts content at the beginning of an HTML element or of each element in a set of HTML elements.
"prev()" on page 923	Returns the previous sibling of the current HTML element.
"remove()" on page 925	Removes an HTML element or a set of HTML elements from the document.
"removeAttr()" on page 926	Removes the specified attribute from the current HTML element.
"removeClass()" on page 926	Removes the specified class from the current HTML element. Has no effect if the class is not present.
"replaceWith()" on page 927	Replaces the current HTML element (with a snippet, for example). Returns the current element.
"show()" on page 928	Shows the HTML element in the template.
"tagName()" on page 928	Returns the HTML tag name of the element, in uppercase. For an example see: "Creating a Table Of Contents" on page 447.

Function	Description
"text()" on page 929	Replaces the text content of the HTML element with the supplied value, or returns the text content of the element if no value is supplied.
"width()" on page 929	Gets or sets the outer width of the HTML element, including padding and borders.

add()

The add() function adds elements to one or more HTML elements, which can be:

- The elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The elements returned by a query in the template (see "query()" on page 845).

add(content)

The add() function adds HTML content to one or more HTML elements and returns the union of this result or result set and other content.

content

A query result. This can be an HTML string or a result set.

Examples

Add one result set to another

This script adds one query result to another and sets the background color to yellow.

```
query("#test1").add(query("#test2")).css("background", "yellow");
```

Note: the way the functions add() and css() are used in this script is called 'chaining'. Chaining is optional; the same could be achieved by storing the results of the queries in a variable:

```
var myResult = query("#test1");
myResult.add(query("#test2");
myResult.css("background", "yellow");
```

Creating an empty result set and adding elements to it

The following script loads snippets in an iteration and adds their elements to an empty result set (using `query()`). Then it replaces a placeholder in the template with the new result.

```
var chapters = query();
for ( var i = 1; i <= 4; i++) {
chapters = chapters.add(loadhtml('snippets/Chapter' + i +
'.html'));
}
results.replaceWith(chapters);
```

Selector	Matched element	Matched element after script execution
#chapters	 {{chapters}}	Chapter 1 Lorem ipsum... Chapter 2 Lorem ipsum... Chapter 3 Lorem ipsum... Chapter 4 Lorem ipsum...

addClass()

Adds the specified class(es) to one ore more HTML elements, which can be:

- The elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The elements returned by a query in the template (see "query()" on page 845).

Adding a class has no effect if the class is already present.

addClass(classname)

Adds the specified class(es) to one or more HTML elements. This has no effect if the class is already present.

classname

String, space separated list of class names.

Examples

This script adds a class name to a paragraph.

```
results.addClass("foo");
```

Selector	Matched element	Matched element after script execution
p	Hello world	Hello world

The following script adds two class names to a paragraph.

```
results.addClass("foo bar");
```

Selector	Matched element	Matched element after script execution
p	Hello world	Hello world

after()

Inserts content after one or more HTML elements, which can be:

- The elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The elements returned by a query in the template (see "query()" on page 845).

See also: "before()" on page 904.

after(content)

Inserts content after one or more HTML elements and creates a new result set.

content

String, HTML string or result set to insert after the matched elements. In case a plain text string is provided, it is automatically wrapped in a element to avoid orphan text nodes to appear in the <body> element.

Examples

This script looks up an element with the ID #salesrep and inserts a paragraph after it.

```
query("#salesrep").after("<p>Lorem ipsum</p>");
```

Matched element	Matched element after script execution
Peter Parker	Peter Parker Lorem ipsum

This script looks up an element with the ID #salesrep, sets its text color to red and inserts a paragraph after it.

```
query("#salesrep").after("<p>Lorem ipsum</p>").css("color","red");
```

Matched element	Matched element after script execution
Peter Parker	Peter Parker Lorem ipsum

Note: the way the functions after() and css() are used in this script is called 'chaining'. Chaining is optional; the same could be achieved by storing the result of the query in a variable:

```
var salesrep = query("#salesrep");
salesrep.after("<p>Lorem ipsum</p>");
salesrep.css("color","red");
```

The following script inserts a paragraph after the elements in the results (the set of HTML elements that match the selector of the script).

```
results.after("<p>Lorem Ipsum</p>");
```

Matched element	Matched element after script execution
Peter Parker	Peter Parker Lorem ipsum

This script looks for the string "Lorem " in the results (the set of HTML elements that match the selector of the script) and inserts the string "ipsum" right after that text. The string is automatically enclosed in a span.

```
results.find("Lorem ").after("ipsum");
```

Matched element	Matched element after script execution
Lorem dolor sit amet, consectetur adipiscing elit.	Lorem ipsum dolor sit amet, consectetur adipiscing elit.

This script looks up an element with the ID #salesrep and inserts a string after it. The string is automatically enclosed in a span.

```
query("#salesrep").after("Lorem Ipsum");
```

Matched element	Matched element after script execution
Peter Parker	Peter Parker Lorem Ipsum

append()

Insert content at the end of one or more HTML elements, which can be:

- The elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The elements returned by a query in the template (see "query()" on page 845).

See also: "prepend()" on page 921.

append(content)

Inserts content as the last element at the end of one or more HTML elements. `Append()` creates a new result set.

content

String, HTML string or result set to insert after the element(s). In case a plain text string is provided, it is automatically wrapped in a element to avoid orphan text nodes to appear in the <body> element.

Examples

This script appends a paragraph to the `results` (the set of HTML elements that match the selector of the script).

```
results.append("<p>Peter Parker</p>");
```

Selector	Matched element	Matched element after script execution
#box	 Personal information 	 Personal information Peter Parker

This script appends a string to the `results` (the HTML elements that match the selector of the script). The string is added to the end of the matched element(s) and wrapped in a Span element.

```
results.append("Peter Parker");
```

Selector	Matched element	Matched element after script execution
.name	 Personal information Name:	 Personal information Name:Peter Parker

Selector	Matched element	Matched element after script execution
	 	

This script's selector is , so the script appends a paragraph to all Div elements in the template.

```
results.append("<p>Peter Parker</p>");
```

Selector	Matched element	Matched element after script execution
div	 Personal information Personal information 	 Personal information Peter Parker Personal information Peter Parker

The following script appends a snippet to a Div element with the ID `box`.

```
var a = loadhtml('snippets/snippet_name.html');
results.append(a);
```

Selector	Matched element	Matched element after script execution
#box	 Personal information 	 Personal information Peter Parker

This script looks for an element with the ID `box` and appends a paragraph to it.

```
query("#box").append("<p>Peter Parker</p>");
```

Matched element	Matched element after script execution
  Personal information 	 Personal information Peter Parker

This script looks for an element with the ID `box`, appends a paragraph to it and colors all text inside the box red.

```
query("#box").append("<p>Peter Parker</p>").css("color","red");
```

Matched element	Matched element after script execution
  Personal information 	 Personal information Peter Parker

Note: the way the functions append() and css() are used in this script is called 'chaining'. Chaining is optional; the same could be achieved by storing the result of the query in a variable:

```
var box = query("#box");
box.append("<p>Peter Parker</p>");
box.css("color","red");
```

attr()

attr(attributeName) : String

Returns the value of the specified attribute of an HTML element which can be:

- The first element in a set of elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The first element in a set of elements returned by a query in the template (see "query()" on page 845).

attributeName

String; the name of the attribute.

Examples

This script - with the selector `img` - stores the source of the first image in a variable.

```
var src = results.attr("src");
```

The following script looks up an image with the ID #image1 and stores its background color in a variable.

```
var imgURL = query("#image1").attr("src");
```

attr(attributeName, value)

Sets the value of the specified attribute of one HTML element or of each element in a result set.

attributeName

String; the name of the attribute.

value

String; value for the attribute.

Examples

This script looks up an image in an element with the ID #calloutbox and sets its alternative text to a value from a data field.

```
var altText = record.fields.FavHobby;
query("#callout img").attr('alt', altText);
```

The following script sets the background color of a specific table cell in an email to red if the value of the field TOTAL has a negative value in the current record.

```
if(record.fields.TOTAL<0) {
        query("#total").attr("bgcolor","red");
}
```

before()

Inserts content before one or more HTML elements, which can be:

- The elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The elements returned by a query in the template (see "query()" on page 845).

See also: "after()" on page 898.

before(content)

Before(content) inserts content before one or more HTML elements and creates a new result set.

content

String, HTML string or result set to insert after the elements. In case a plain text string is provided, it is automatically wrapped in a element to avoid orphan text nodes to appear in the <body> element.

Examples

This script looks for an element with the ID `salesrep`and inserts a paragraph before that element.

```
results.before("<p>Lorem Ipsum</p>");
```

Selector	Matched element	Matched element after script execution
#salesrep	Peter Parker	Lorem ipsum Peter Parker

This script does the same, but it uses the query() function to look up the element.

```
query("#salesrep").before("<p>Lorem ipsum</p>");
```

Matched element	Matched element after script execution
Peter Parker	Lorem ipsum Peter Parker

The following script looks for an element with the ID salesrep, inserts a paragraph before that element and colors that element red.

```
query("#salesrep").before("<p>Lorem ipsum</p>").css("color","red");
```

Matched element	Matched element after script execution
Peter Parker	Lorem ipsum Peter Parker

Note: the way the functions `before()` and `css()` are used in this script is called 'chaining'. Chaining is optional; the same could be achieved by storing the result of the query in a variable:

```
var salesrep = query("#salesrep");
salesrep.before("<p>Lorem ipsum</p>");
salesrep.css("color","red");
```

The following script searches the results for the string "ipsum" and puts "Lorem " before it. "Lorem " is automatically wrapped in a Span element.

```
results.find("ipsum").before("Lorem ");
```

Matched element	Matched element after script execution
ipsum dolor sit amet, consectetur adipiscing elit.	Lorem ipsum dolor sit amet, consectetur adipiscing elit.

The following script looks for an element with the ID salesrep and inserts the text "Lorem Ipsum" before that element. "Lorem Ipsum" is automatically wrapped in a Span element.

```
query("#salesrep").before("Lorem Ipsum");
```

Matched element	Matched element after script execution
ipsum dolor sit amet, consectetur adipiscing elit.	Lorem Ipsum Peter Parker

children()

Returns the immediate children (inner HTML) of one or more HTML elements, which can be:

- The elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The elements returned by a query in the template (see "query()" on page 845).

Examples

This script retrieves the inner HTML of an element selected from a snippet.

```
var snippet = loadhtml('snippets/snippet.html','#foobar').children
();
results.append(snippet);
```

The following script retrieves the inner HTML of the elements and then performs a find/replace.

```
var snippet = loadhtml('snippets/snippet.html','#foobar').children
();
snippet.find('@firstname@').text('foobar');
results.append(snippet);
```

clone()

This function returns a copy of one HTML element or of a set of HTML elements, which can be:

- The elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The elements returned by a query in the template (see "query()" on page 845).

See also: "Dynamically adding sections (cloning)" on page 440.

To duplicate an existing template element, clone it before calling `append()`; see "append()" on page 900.

Examples

This script performs an iteration over the elements in the `results` (the elements that match the selector of the script).

```
var row = query("tbody tr", results).clone();
query("tbody", results).append(row);
```

The following script clones an existing table row to match the number of rows in a detail table. Afterwards it iterates over the rows to populate the fields.

```
// Create the number of rows based on the records in the detail
table
// We start at 1 so the boilerplate row is used too and there is no
need to delete that row
for(var r = 1; r < record.tables['detail'].length; r++) {
results.parent().append(results.clone());
}

// Iterate over the rows and populate them with the data from the
accompanying data row
query("#table_2 > tbody > tr").each(function(i) {
this.find('@ItemNumber@').text( record.tables['detail'][i].fields
["ItemNumber"]);
this.find('@ItemOrdered@').text( record.tables['detail'][i].fields
["ItemOrdered"]);
this.find('@ItemTotal@').text( record.tables['detail'][i].fields
["ItemTotal"]);
this.find('@ItemDesc@').text( record.tables['detail'][i].fields
["ItemDesc"]);
this.find('@nr@').text(i);
});
```

The following script clones and populates a boilerplate row. Once completed you will need to hide the boilerplate row.

closest()

This function gets the first parent element that matches a selector, by testing the element itself and traversing up through its ancestors in the DOM tree. (In HTML, a parent is an element that contains another element.)

The function can be used for:

- The elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The elements returned by a query in the template (see "query()" on page 845).

To get a child element or all child elements, use `children()` (see "children()" on page 907).

The `closest()` command is based on the `closest()` command found in the jQuery library: https://api.jquery.com/closest/.

To get an element's sibling element, you can use "prev()" on page 923 or "next()" on page 916.

closest(selector)

For one HTML element or for each element in a set, this function gets the first element that matches the selector by testing the element itself and traversing up through its ancestors in the DOM tree.

selector

A String containing an HTML tag (without the angle brackets, <>).

Examples

The following script looks up all table rows in the template that contain an <input> element.

```
query("input").closest("tr");
```

This code gets the closest 'parent' row for each element that matches the selector of the script (collected in the `results` object):

```
results.closest("tr");
```

The rows could be colored red within the same statement:

```
results.closest("tr").css('background-color','red');
```

css()

Gets the value of a style property of one HTML element, or sets one or more CSS properties of one or more HTML elements.

css(styleName) : String

Returns the value of the specified CSS property of an HTML element, which can be:

- The first element in a set of elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The first element in a set of elements returned by a query in the template (see "query()" on page 845).

propertyName
String; the name of the CSS property.

Examples

This script stores the text color of the `results` (the HTML elements that match the selector of the script) in a variable.

```
var textcolor = results.css("color");
```

The following script looks up an element with the ID #calloutbox and stores its background color in a variable.

```
var backgroundcolor = query("#calloutbox").css("background-color");
```

css(styleName, value)

Function to set a CSS property of one HTML element or of each element in a result set.

propertyName
String; the name of the CSS property.

value
String; value for the CSS property or a map of property-value pairs to set.

Examples

This script looks up an element with the ID #calloutbox and sets its text color to red.

```
query("#callout p").css('color' , 'red');
```

The following script does the same, but it only sets the text color to red if in the current record the value of the field 'accounttype' is 'PRO'.

```
if(record.fields.accounttype == "PRO") {
query("#callout p").css("color","red");
}
```

This script sets the text color of the results to a hexadecimal color code.

```
results.css('color' , '#669900');
```

This script loads a snippet into a variable. Then it finds/replaces text in the snippet and applies a css property to the replacing text.

```
var mysnippet = loadhtml('snippets/snippet vars.html');
mysnippet.find('@var@').text('OL Connect').css('text-
decoration','underline');
results.replaceWith(mysnippet);
```

css(properties)

Function to set one or multiple CSS properties of one or more HTML elements, which can be:

- The elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The elements returned by a query in the template (see "query()" on page 845).

properties

Array; map of property-value pairs to set.

Examples

This script colors the text of the `results` (the set of HTML elements that match the selector of the script) red and makes it bold.

```
results.css({'color' : 'red', 'font-weight' : 'bold'});
```

empty()

Removes the contents (child elements and inner HTML) from one or more HTML elements, which can be:

- The elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The elements returned by a query in the template (see "query()" on page 845).

Use "remove()" on page 925 to remove the elements themselves.

Example

This script empties all Span elements found in the template.

```
results.empty();
```

Selector	Paragraph before script execution	Paragraph after script execution
span	Lorem ipsum dolor sit amet, consectetuer adipiscing elit.	Lorem ipsum amet, consectetuer adipiscing elit.

find()

find(textToFind)

Performs a deep search for textToFind in the children of each element, and returns a new result set with elements that surround the occurrences.

textToFind

A String that contains the search text.

Example

The following piece of code loads a snippet, then looks for placeholders using find(), and replaces them with a text.

```
var mysnippet = loadhtml('snippets/snippet.html');
mysnippet.find('@var1@').text('OL Connect 1');
mysnippet.find('@var2@').html('<i>OL Connect 2</i>').css('text-
decoration','underline');
results.replaceWith(mysnippet);
```

hasClass()

hasClass(classname) : Boolean

Returns true if the HTML element or the first element in a result set has the specified class.

classname

String containing one class name.

Example

This script checks if the first of the `results` (the set of elements matching the selector of the script) has the class 'green'. If so, it colors the text of all the elements in the `results` green.

```
if (results.hasClass('green')) {
        results.css('color', 'green');
}
```

height()

Gets or sets the outer height, including padding and borders, of an HTML element, or of the first element in a result set (see "results" on page 972), i.e. the set of HTML elements that match the selector of the script or of another query in the template (see "query()" on page 845).

height(): Number

Returns the outer height of this element, including padding and borders, excluding margins. To include margins, call `height(true)`.

The returned value is the height in pixels, without measurement unit (e.g. 400).

height(value): void

Sets the outer height of this element, including padding and borders, excluding margins. To include margins, call `height(value, true)`.

value

Number. Give the height in pixels, but without the measurement unit (e.g. 400, not 400px).

Examples

This script adds 20 pixels to the width and height of an element.

```
var h = results.height();
var w = results.width();

h = h + 20;
w = w + 20;

results.height( h );
results.width( w );
```

The following script does the same, but in this case, margins are included.

```
var h = results.height(true);
var w = results.width(true);

h = h + 20;
w = w + 20;

results.height( h, true);
results.width( w, true);
```

hide()

Hides one or more HTML elements, which can be:

- The elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The elements returned by a query in the template (see "query()" on page 845).

This doesn't remove the elements; to make them visible again, use the function "show()" on page 928.

These functions are used by the Conditional Script Wizard, as you can see when you open a Conditional Script and click the **Expand** button; see "Showing content conditionally" on page 358.

Example

This script hides or shows the elements matched by the selector of the script (which are stored in the `results` object), depending on the value of the data field `Country` in the current record.

```
if (record.fields["Country"] == "CANADA") {
results.show();
} else {
results.hide();
}
```

html()

html() : String

Returns the inner HTML of an HTML element, which can be:

- The first element in a set of elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The first element in a set of elements returned by a query in the template (see "query()" on page 845).

html(value)

Replaces the inner HTML of an HTML element or of each element in a result set with the supplied value.

value

A String that may contain HTML tags.

Examples

The following script loads part of a snippet based on the value of a field, and then inserts the content into the document using `html()`.

```
var promoTxt = loadhtml('snippets/promo-en.html', '#' +
record.fields['YOGA']);
results.html(promoTxt);
```

The following script loads a snippet. Then it looks for a placeholder (@var2@) in the text of that snippet and replaces every found placeholder by the text 'OL Connect 1'. It uses html() so the HTML formatting (and) will indeed be interpreted as HTML. Finally, it places the snippet in the template.

```
var mysnippet = loadhtml('snippets/snippet.html');
mysnippet.find('@var1@').html('<i>OL Connect 1</i>');
results.replaceWith(mysnippet);
```

info()

Returns pagination information for one HTML element or for the first element in a result set (see "results" on page 972) of a query across all sections in a Print context (see "query(selector)" on page 971).

The returned information is of the type `PaginationInfo` (see "PaginationInfo" on page 971).

This function can only be used in a Post Pagination Script ; see "Post Pagination Script API" on page 967.

For an example see: "Creating a Table Of Contents" on page 447.

next()

`next()` returns the next sibling of an HTML element, which can be:

- The elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The elements returned by a query in the template (see "query()" on page 845).

In HTML, siblings are HTML elements that share the same parent, i.e. HTML elements in the same container element.
Note that a sibling can be a different type of element. For example, if a paragraph, an image and a table follow each other in the same Box, they are siblings.

> **Tip**
>
> To get the **previous** sibling, use previous() (see "prev()" on page 923).

What if there are no siblings?

If you call `prev()` or `next()` on a **collection** of elements (either "results" on page 972 or the result set of "query()" on page 845), the function will return a collection of siblings. If none of the elements has a previous or next sibling, the collection will be empty. Actions performed on an empty collection will simply have no effect.

If you call `prev()` or `next()` on a **single** element (this in an 'Each matched element' script), and the element has no previous or next sibling, the function returns `null`. Performing an operation on a `null` value will result in an error, which means that the rest of the script won't be executed. In this case, it is useful to check the return value, unless it's certain that it will never be `null`.

Example

Assume that there are three consecutive paragraphs in a Box and that the second of those paragraphs has an ID that matches the selector of this script. The paragraph is stored in the results object (see "results" on page 972). The following script changes the font-weight of the next paragraph - the third paragraph in the Box - to bold.

```
results.next().css("font-weight", "bold");
```

If the elements in results don't have siblings, this line of code will have no effect. It also won't result in an error.

In an 'Each matched element' script you would first need to check the return value of this.next():

```
var nextSibling = this.next();
if nextSibling != null { nextSibling.css("font-weight", "bold"); }
```

overflows()

The `overflows()` method returns a boolean value indicating whether an HTML element overflows its box boundaries.

This function could for instance be used in a design where content needs to flow separately from the main text flow, or where a new, full-page, multi-column box should be inserted when the current one overflows.

Note that when called in a Standard script, this method runs *before* the pagination of Print output. After pagination - in the output, and in Preview mode - the element's overflow may have changed, especially when the element is combined with floating elements (i.e. elements that have the CSS style float) and located near a page-break.
To prevent this from happening, it may help to set the element's CSS style to `overflow:hidden`.
The `overflows()` method can also be used in a Post Pagination script.

> **Tip**
>
> You don't need a script to resize text in order to make it fit in a box. The Copy Fit feature automatically scales text to the available space (see "Copy Fit" on page 311).

Examples

```
var $box = query("#mybox");
while ( ! $box.overflows()) {
        //do something
}
```

A slightly shorter version:

```
while ( ! query("#mybox").overflows()) {
        //do something
}
```

The following script selects three boxes that all have the class box (selector: .box). It loops over them, inserting as many products from an array as possible into the box. In fact, it inserts one too many and removes the last one before moving on to the next box.

```
var products =
["Appetizer","Beans","Beef","Lettuce","Sprouts","Coconut","Juice","
Soup","Coriander","Cheese","Pasta","Sugar","Vinegar","Bread"];
var i = 0;
```

```
results.each( function() {
        // Add elements to the box until it overflows.
        while ( ! this.overflows() && i < products.length ) {
                this.append("<p>" + products[i] + "</p>");
                i++;
        }
        // Go one step back unless we processed all items in our array.
        if( i < products.length ) {
                query("p:last-child", this).remove();
                i--;
        }
});
```

pageref()

Returns a **marker** that will be replaced with the element's page number after pagination. This only works for elements in the section that is currently being merged.

Example

Creating a table of contents

The following script creates a table of contents for all level 1 headings (<h1> elements) with the class `title` in one section.

```
var toc = '<ul ID="toc">';
query('h1.title').each(function() {
 toc += '<li>' + this.text() + ' <span class="li_toc">' +
this.pageref() + '</span></li>';
});
toc += '</ul>';
results.after(toc);
```

The first line creates a variable for the table of contents, which will be a list (a <ul> element with the ID `toc`). The start tag of the list is added to the variable.
The next line does a query for all level 1 headings (<h1> elements) with the class `title` in the current section. With `each()` the script loops through them. For each of the headings it adds a line item to the list, with the text (`this.text()`) and the page reference of the respective heading.
After the loop, the end tag of the list is added to the variable.
Finally, the script adds the variable - that now contains the table of contents - after the `results`.

The `results` object contains the elements that match the selector of the script. So, if the script's selector selects the title of the table of contents, the table of contents will be added after that.

The following style rules, added to the style sheet, will align the chapter titles to the left and the page numbers to the right:

```
#toc li {
 text-align:left;
}
#toc span {
 float: right;
}
```

Note that these styles use the list's ID, that was defined in the first line of code. For information about style sheets, see "Styling templates with CSS files" on page 296.

parent()

Returns the parent(s) of one or more HTML elements, which can be:

- The elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The elements returned by a query in the template (see "query()" on page 845).

In HTML, a parent is an element that contains another element.
To get an ancestor that matches a particular selector, use `closest()` (see "closest()" on page 908).

To get an element's sibling element, you can use "prev()" on page 923 or "next()" on page 916.

Example

Assume that there are three paragraphs in a Box and that one of those paragraphs matches the selector of this script. The paragraph is stored in the `results` object (see "results" on page 972). The following script retrieves the Box (which is the parent of the paragraph) using `results.parent()`, and then changes its background color to red.

```
results.parent().css('background-color' , 'red');
```

prepend()

Insert content at the beginning of one or more HTML elements, which can be:

- The elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The elements returned by a query in the template (see "query()" on page 845).

See also: "append()" on page 900.

prepend(content)

Insert content as the first element to each element in the set of HTML elements that match the selector of the script or of another query in the template (see "query()" on page 845). `Append` creates a new result set.

content

HTML string, string or HTML string to insert after the matched elements. In case a plain text string is provided, it is automatically wrapped in a element to avoid orphan text nodes to appear in the <body> element.

Examples

This script inserts a heading as the first element in an element that has the ID `#box`.

```
results.prepend("<h1>Personal information</h1>");
```

Selector	Matched element	Matched element after script execution
#box	 Peter Parker 	 Personal information Peter Parker

This script inserts a heading as the first element in an element that has the class `name`.

```
results.prepend("<b>Name: </b>");
```

Selector	Matched element	Matched element after script execution
.name	 Personal information Peter Parker 	 Personal information Name: Peter Parker

This script inserts content in multiple elements at the same time.

```
results.prepend("<h1>Personal information</h1>");
```

Selector	Matched element	Matched element after script execution
div	 Peter Parker Peter Parker 	 Personal information Peter Parker Personal information Peter Parker

This script prepends a snippet that contains the text "<h1>Personal information</h1>".

```
var a = loadhtml('snippets/snippet.html');
results.prepend(a);
```

Selector	Matched element	Matched element after script execution
div	 Peter Parker 	 Personal information Peter Parker

This script uses the function query() to find a box. Then it inserts a heading as the first element in that box.

```
query("#box").prepend("<h1>Personal information</h1>");
```

Matched element	Matched element after script execution
  Peter Parker 	 Personal information Peter Parker

This script uses the function query() to find a box, prepends a heading and sets the text color of the entire box to red.

```
query("#box").prepend("<h1>Personal information</h1>").css
("color","red");
```

Matched element	Matched element after script execution
  Peter Parker 	 Personal information Peter Parker

Note: the way the functions prepend() and css() are used in this script is called 'chaining'. Chaining is optional; the same could be achieved by storing the result of the query in a variable:

```
var box = query("#box");
box.prepend("<p>Peter Parker</p>");
box.css("color","red");
```

prev()

`prev()` returns the previous sibling of an HTML element, which can be:

- The elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The elements returned by a query in the template (see "query()" on page 845).

In HTML, siblings are HTML elements that share the same parent, i.e. HTML elements in the same container element.
Note that a sibling can be a different type of element. For example, if a paragraph, an image and a table follow each other in the same Box, they are siblings.

> **Tip**
>
> To get the **next** sibling, use `next()` (see "next()" on page 916).

What if there are no siblings?

If you call `prev()` or `next()` on a **collection** of elements (either "results" on page 972 or the result set of "query()" on page 845), the function will return a collection of siblings. If none of the elements has a previous or next sibling, the collection will be empty. Actions performed on an empty collection will simply have no effect.

If you call `prev()` or `next()` on a **single** element (this in an 'Each matched element' script), and the element has no previous or next sibling, the function returns `null`. Performing an operation on a `null` value will result in an error, which means that the rest of the script won't be executed. In this case, it is useful to check the return value, unless it's certain that it will never be `null`.

Example

Assume that there are three consecutive paragraphs in a Box and that the second of those paragraphs has an ID that matches the selector of this script. The paragraph is stored in the results object (see "results" on page 972). The following script changes the font-weight of the previous paragraph - the first paragraph in the Box - to bold.

```
results.prev().css("font-weight", "bold");
```

If the elements in results don't have siblings, this line of code will have no effect. It also won't result in an error.

In an 'Each matched element' script you would first need to check the return value of `this.prev()`:

```
var prevSibling = this.prev();
if prevSibling != null { prevSibling.css("font-weight", "bold"); }
```

remove()

Removes the current element or each element in a set from the DOM.

This function returns a new result set containing each removed element. These can be changed and inserted in the document. This could be beneficial in terms of performance, as manipulating elements inside the DOM is relatively time consuming.

Examples

This script removes all Span elements found in the template.

```
results.remove();
```

Selector	Paragraph before script execution	Paragraph after script execution
span	Lorem ipsum dolor sit amet, consectetuer adipiscing elit.	Lorem ipsum amet, consectetuer adipiscing elit.

The selector of the following sample script is `tbody`. Before this script runs, the table body consists of a single placeholder row with three cells. After running the script, it contains thirty rows. To improve performance, most of the DOM manipulation takes place on detached elements.

```
// Detach the placeholder row from the DOM
var row = query("tr", results).remove();
```

```
// Modify the cells of this row
var cells = row.children();
cells[0].html("some text").css("background-color", "yellow");
cells[1].html("some text").css("font-weight", "bold");
cells[2].html("some text");

// Create a number of copies
var rows = row.clone();
for (var i = 0; i < 30; i++) {
       rows = rows.add(row.clone());
}

// Attach all copies to the DOM as children of tbody
results.append(rows);
```

removeAttr()

Removes the specified HTML attribute from an element or from each element in a set of elements. To add or change an attribute, use `attr()` (see "attr()" on page 903).

removeAttr(attributeName)

attributeName

String; the name of the attribute.

Examples

This script looks up an email field in a form (which is an <input> with the ID #email1) and removes its `readonly` attribute.

```
query("#email1").removeAttr('readonly');
```

removeClass()

Removes the specified class from the current HTML element or from each element in a result set. This has no effect if the class is not present.

removeClass(classname)

classname

String, space separated list of class names.

Examples

This script removes the class name "foo" from all elements in the results that have this class.

```
results.removeClass("foo");
```

Selector	Matched element	Matched element after script execution
p	Hello world	Hello world

replaceWith()

Replaces one HTML element or each element in a set of HTML elements.

replaceWith(content)

Replaces one HTML element or each element in a set of HTML elements. Returns the result set.

content

A query result. This can be an HTML string or a result set.

Examples

Replace elements with a snippet

The following script loads a snippet and then replaces the elements matched by the script's selector with the snippet.

```
var snippet = loadhtml('snippets/mysnippet.html');
results.replaceWith(snippet);
```

Replace elements with a set of snippets

The following script loads snippets and adds their elements to a new, empty result set (using `query()`). Then it replaces a placeholder in the template with the set of snippets.

```
var chapters = query();
for ( var i = 1; i <= 4; i++) {
chapters = chapters.add(loadhtml('snippets/Chapter' + i +
'.html'));
```

```
}
results.replaceWith(chapters);
```

show()

Shows one or more HTML elements, which can be:

- The elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The elements returned by a query in the template (see "query()" on page 845).

To hide elements (again), use the function "hide()" on page 914.

These functions are used by the Conditional Script Wizard, as you can see when you open a Conditional Script and click the **Expand** button; see "Showing content conditionally" on page 358.

Example

This script hides or shows the elements matched by the selector of the script (which are stored in the `results` object), depending on the value of the data field `Country` in the current record.

```
if (record.fields["Country"] == "CANADA") {
results.show();
} else {
results.hide();
}
```

tagName()

Returns the HTML tag name of one HTML element in uppercase (e.g. H1, P, ARTICLE).

Example

This Post Pagination Script finds level 1 and 2 headings in all Print sections and stores their page number, text and HTML tag name.

```
merge.context.query("h1, h2").each(function() {
        var pageNo = this.info().pageNo;
        var text = this.text();
        var level = this.tagName();
});
```

This code is used in a script that inserts a table of contents. For the full script and explanation see "Creating a Table Of Contents" on page 447.

text()

text() : String

Returns the text content of an HTML element or of the first element in a result set (see "results" on page 972).

Example

This script loads a snippet into a variable and retrieves an element from the snippet using `query()` and `text()`.

```
var mysnippet = loadhtml('snippets/text-root-wrapped.html');
var subject = query("#subject", mysnippet).text();
results.append("<p style='font-weight: bold;'>" + subject +
"</p>");
```

text(value)

Replaces the text content of one HTML element or of each element in a result set with the supplied value.

Example

This script loads a snippet, then looks for placeholders using find(), and replaces them using text(value).

```
var mysnippet = loadhtml('snippets/snippet.html');
mysnippet.find('@var1@').text('OL Connect 1');
mysnippet.find('@var2@').html('<i>OL Connect 2</i>').css('text-
decoration','underline');
results.replaceWith(mysnippet);
```

width()

Gets or sets the outer width, including padding and borders, of an HTML element or of the first element in result set (see "results" on page 972), i.e. the set of HTML elements that match the selector of the script or of another query in the template (see "query()" on page 845).

width(): Number

Returns the outer width of this element, including padding and borders, excluding margins. To include margins, call `width(true)`.

The returned value is the width in pixels, without measurement unit (e.g. 400).

width(value): void

Sets the outer width of this element, including padding and borders, excluding margins. To include margins, call `width(value, true)`.

value

Number. Give the width in pixels, but without the measurement unit (e.g. 400, not 400px).

Examples

This script adds 20 pixels to the width and height of an element.

```
var h = results.height();
var w = results.width();

h = h + 20;
w = w + 20;

results.height( h );
results.width( w );
```

The following script does the same, but in this case, margins are included.

```
var h = results.height(true);
var w = results.width(true);

h = h + 20;
w = w + 20;

results.height( h, true);
results.width( w, true);
```

translate()

When a template has been prepared to use the translation feature (see "Translating templates" on page 467), this feature is also accessible via a template script.

The `translate()` function gets the translation for a message, provided that a translation entry can be found for it (with the same context, if given).

If the text to be translated does not already exist in the template, you can create a translation entry for it by using the **Add Empty String** button on the "Translations pane" on page 594.

Note that calling the `translate()` function may be unnecessary if a script adds text to an element that has the `data-translate` attribute; see "Tagging text that is inserted by a script" on page 471. In that case, any existing translation will be applied automatically.

translate(message, context)

The `translate()` function gets the translation of a source text with an (optional) context. It returns the message as-is if no translation is found.

message

String containing the message (the source text).

context

The message's context (optional). A context may be specified when creating a translation entry. Specifying a context makes it possible to translate a certain text in different ways.

Example

The following line of code gets the translation of the string "Submit" that was registered with the context "button".

```
var str = translate( "Submit", "button" );
```

Control Script API

The table below lists the objects that are the most important in Control Scripts. Click through to the object to find a description and sample scripts.
See "Control Scripts" on page 429 for information about this kind of scripts, how to insert them and what you can do with them.

Other objects that are available to Control Scripts

The list below isn't exhaustive: most of the objects listed in the Designer API (see "Standard Script API" on page 810) are also available in Control Scripts. Not all of those objects can be

used in Control Scripts, however. This is because Control Scripts differ from template scripts (see "What Control Scripts are" on page 430). For example, the `query()` function can't be used in a Control Script because it requires access to the DOM which a Control Script doesn't have.

Object	Usage
channel (see "Channel" on page 964)	The channel for which output is generated. This is registered in the `merge` object: `merge.channel`.Note that the channel doesn't change when the output consists of different contexts. When generating email, for example, the channel is EMAIL, even when merging the Print context to attach it to the email.
"context" on page 969	Object that contains one context and its sections. It is accessed through the `template` object: `merge.template.contexts`. To get access to one context, you have to specify the ContextType (see "ContextType" on page 965), for example: `var printContext = merge.template.contexts.PRINT;`. Through the `merge` object you can find out which context is currently being merged: `merge.context`.
"masterpage" on page 934	The `masterpage` object is used to set a Master Page's header and footer or to replace its entire HTML body.
"merge" on page 989	The `merge` object gives access to the template with all of its contexts and sections .
"record" on page 848	The current record in the main data set. To get the value of a field in the record, use `record.fields['fieldname']` or `record.fields.fieldname`.
"section" on page 978	Much of the Control Script magic is performed by setting one of the fields of the `section` object. Via the `section` object you can omit, select and clone sections; add a background to a Print section; add a header to an email; etc. A section can be retrieved via the context that it belongs to, using `merge.template.contexts.ContextType.sections["section name"]`. For example: `merge.template.contexts.PRINT.sections["Section`

Object	Usage
	`EN"].`
"template" on page 960	The `template` object contains all contexts and sections. It is accessed through the `merge` object: `merge.template`.

context

In a Control Script, the `context` object represents one context in the template.

Which contexts are available in the template can be queried using `merge.template.contexts`.

The context being merged can be queried using `merge.context`.

Field	Type	Description
sections	Array	Array of sections (see "section" on page 978) inside a particular context defined in the template. **Note**: When using `merge.context.sections` keep in mind that for example 'Section X' might only exist in your Print context, so using `merge.context.sections['Section X']` without enclosing it in the `if` statement `if (merge.context.type == ContextType.PRINT) {}` will yield an error when the script runs for other contexts. Alternatively, use the `template` object to access a specific context: `merge.template.contexts.PRINT.sections['Section X']`.
type	"ContextType" on page 965	The context type: PRINT, EMAIL or WEB.

Function	Description
"query	Runs a query across all sections in the Print context. This function is only

Function	Description
(selector)" on page 971	available in Post Pagination Scripts, which are only applied to the Print context. See "Post Pagination Scripts" on page 445.

Example

This script checks if the output channel is EMAIL and if the context to be merged is the Print context (which happens if the Print context is attached to an email). If this is the case, it includes and excludes certain Print sections from the output.

```
if (channel == Channel.EMAIL) {
        if (merge.context.type == ContextType.PRINT) {
                merge.context.sections['Section 1'].enabled = false;
                merge.context.sections['Section 2'].enabled = false;
                merge.context.sections['Section 3'].enabled = true;
        }
}
```

masterpage

The `masterpage` object is used to set a Master Page's header and footer or to replace its entire HTML body. It can **only** be accessed in Control Scripts (see "Control Scripts" on page 429 and "Control Script API" on page 931). Trying to access it in another type of script will result in an error.
Note that Master Pages are only used in a Print context (see "Master Pages" on page 156).

In script, Master Pages are retrieved via the masterpages Array which contains all Master Pages in the current template (see "template" on page 960).
For example: `var myMasterpage = merge.template.masterpages["Master page 1"];`.

Functions and fields

Field	Type	Description
"html()" on page 958		Gets or sets the HTML of the body element.
"margins" on page 987	Margins	Allows to set the header and footer of a Master Page, using a Measurement string, for example: "2in" (this sets a margin to

Field	Type	Description
		two inches).

html()

In a Control Script, the `html()` function of a section or Master Page can be used to get the initial contents of its <body>, and modify them. This makes it possible, for example, to populate a section or Master Page with elements retrieved from a Content Management system, before Standard Scripts run.

html()

Gets the initial contents of the <body> of a section or Master Page.

html(value)

Replaces the contents of the <body> of a section or Master Page with the supplied value. This function is only available in Control Scripts. See also: "section" on page 978 and "masterpage" on the previous page.

value

A String that may contain HTML tags.

Examples

The following script uses `html()` to retrieve the contents of a section and add a paragraph to it.

```
var foo = merge.context.sections["Section 1"].html();
foo += "<p>hello world</p>";
merge.context.sections["Section 1"].html( foo );
```

The following script loads a snippet based on the value of a field, and then replaces the content of a Print section with the snippet using `html()`.

```
var mySection = merge.context.sections["Section 1"];
var promoTxt = loadhtml('snippets/promo-' + record.fields['City'] +
'.html');
mySection.html(promoTxt);
```

margins

The `margins` object is used to set a Print section or Master Page's margins in a Control Script (see "Control Scripts" on page 429 and "Control Script API" on page 931).
Note that Master Pages are only used in a Print context (see "Master Pages" on page 156).

The `margins` object is retrieved via the `section` object (see "section" on page 978) or via the `masterpages` array in a template (see "masterpage" on page 934), respectively.

Fields

Field	Type	Description
bottom **left** **right** **top**	String	These fields allow to set the bottom, left, right and top of a Print section, using a Measurement string, for example: "2in" (this sets a margin to two inches).
footer **header**	String	These fields allow to set the header and footer of a Master Page, using a Measurement string.

Examples

Setting the margins of a Print section

```
var sectionA = merge.template.contexts.PRINT.sections["Section A"];
sectionA.margins.top = "2in";
sectionA.margins.left = "2in";
sectionA.margins.right = "2in";
sectionA.margins.bottom = "2in";
```

Setting the header and footer of a Master Page

```
var masterA = merge.template.masterpages["Master page A"];
masterA.margins.header = "2in";
masterA.margins.footer = "2in";
```

media

The media object can be used to specify, enable and position a stationery's front and back in a Control Script (see "Control Scripts" on page 429 and "Control Script API" on page 931).
Note that Media are only used in Print sections (see "Media" on page 159).

The available media are listed in, and retrieved via, the template object (see "template" on page 960), for example: `var myMedia = merge.template.media.My_Media`.

Fields

Field	Type	Description
height	String	The height of the Media using a Measurement string. For example, "11in" is 11 inches. In a Control Script this property can be used to change the height of a Media. In other scripts it is only possible to get the height of the Media (read-only).
stationery	Stationery	The Stationery's object's `front` and `back` fields are used to set the front and the back of a Media; see "front, back" below.
width	String	The width of the Media using a Measurement string. For example, "8.5in" is 8.5 inches. In a Control Script this property can be used to change the width of a Media. In other scripts it is only possible to get the width of the Media (read-only).

front, back

The `front` and `back` fields of the Stationery object are used to set the front and the back of a Media (see "media" above).
Both `front` and `back` have the following fields.

Field	Type	Description
enabled	boolean	When enabled, the stationery will be included in the output (Print sections only).
url	String	Specifies the image to use as virtual stationery. For a file named My_Media.pdf, stored inside the template resources, the url would be `images/My_Media.pdf`. The complete syntax for an external file is: **file:///<host>/<path>**. If the host is "localhost", it can be omitted, resulting in **file:///<path>**, for example: `file:///c:/users/Adminstrator/Pictures/My_Media.jpg`.
page	Number	PDF and TIFF files may count more than one page. Specify the page number to use.
position	"MediaPosition" on page 966	The `position` is used to place the stationery on the page (absolute, centered, fit to media).
top	Measurement	The vertical offset from the top of the page, used to position the stationery (only when absolute positioning is selected). This value can be negative.
left	Measurement	The horizontal offset from the left of the page, used to position the stationery (only when absolute positioning is selected). This value can be negative.

merge

In Control Scripts, the root level instance of the object `merge` is the entry point from where you can query and change the way contexts are merged. It gives access to the template with all its contexts and sections.
For more information about Control Scripts, see "Control Scripts" on page 429 and "Control Script API" on page 931.
Some of the objects are also useful in Post Pagination Scripts; see "Post Pagination Scripts" on page 445 and "Post Pagination Script API" on page 967.
For sample scripts, follow the links to the respective objects.

Field	Type	Description
channel	"Channel" on page 964	The final output channel: EMAIL, PRINT or WEB. The channel doesn't change when the output consists of different contexts. When generating an email, for example, the channel is EMAIL, even when merging the Print context to attach it to the email.
"context" on page 969	Context	The context rendered by this merge run. If for one record, different contexts need to be output (for example, when the Print context is attached to an email) a record is merged multiple times: once per context. Per merge run, `merge.context` shows with which context the record is merged.
pagination	"Pagination" on page 991	Contains the total page count and sheet count of all sections in the Print context after pagination.
"section" on page 978	Section	In Standard Scripts, this object defines the section that is being merged. **Note!** In Control Scripts, `merge.section` is not available. To retrieve any section in a Control Script, use: `merge.template.contexts.ContextType.Section['Section name'];` (for example: `merge.template.contexts.PRINT.sections["Section EN"]`). In Post Pagination Scripts, only Print sections are available.
"template" on page 960	Template	This object contains the template and all of its contexts. It can be used to find out which contexts are available in the template, using `merge.template.contexts` (see "context" on page 969) and to manipulate the sections in those

Field	Type	Description
		contexts (see "section" on page 978).

resource()

The `resource()` function returns information about an image resource. This function is useful in a Control Script, for example to check the number of pages or the page height and width before setting it as a background (see "Control Script: Setting a Print section's background" on page 438).

This function can also be used to check if a file exists. It returns `null` if the file does not exist. There is no need to explicitly check for `null`; a check to see if a resource exists can simply be written as:

```
var info = resource(path);
if (!info) {
// File does not exist
}
```

resource(location, pageNumber)

location

The location should be a URL relative to the template root or an absolute file-based URL (without protocol), e.g. "C:/myfile.pdf".

pageNumber (optional)

The desired page. Counting starts at 1. If no page number is given, information about the first page will be retrieved.

The returned object is of the type ImageInfo. It has the following fields:

Field	Type	Description
height	float	The height of the current page (in points).
page	Number	Current page number (counting from 1) within the resource.

Field	Type	Description
pages	Number	The total number of pages in the resource.
permissions	"Permissions" below	PDF only. Allows to verify if a PDF is password-protected or has restrictions for printing.
width	float	The width of the current page (in points)

Examples

This script retrieves the second page of a PDF that is present in the template's resources.

```
var pdf = resource("images/stamp.pdf", 2);
var height = pdf.height;
var width = pdf.width;
var numberOfPages = pdf.pages;
```

In the following script, the function is used to check if a file exists.

```
if(resource("C:/paw.pdf")){
        //exists
} else {
        //oops
}
```

Permissions

Allows to verify if a PDF is password-protected or has restrictions for printing.
This is part of the information about an image that is returned by the `resource()` function; see "resource()" on the previous page).

> **Note**
>
> If a PDF that does not allow content copying, Connect cannot handle the PDF at all and we cannot determine the permissions. In that case `hasPassword` will be `true` (even if the password is not set) and `printingAllowed`will be `unknown`.

Field	Type	Description
hasPassword	Boolean	Will be `true` if the resource is a password protected PDF.
printingAllowed	String	Will be either "highres" (full permissions), "lowres", "none", or "unknown".

Example

The following script logs the permissions for a PDF.

```
var pdf = resource("images/myPDF.pdf");
var willPrint = pdf.permissions.printingAllowed;
var hasPW = pdf.permissions.hasPassword;
logger.info('Printing restrictions: ' + willPrint);
logger.info('Password-protected: ' + hasPW);
```

section

The section object can be used to query and modify how the section (and the related context) will be outputted. It is one of the most important objects in Control Scripts (see "Control Scripts" on page 429 and "Control Script API" on page 931).

Retrieving a section

A section can be retrieved using `merge.template.contexts.ContextType.sections["section name"]`, for example: `merge.template.contexts.PRINT.sections["Section EN"]`.

A section can also be retrieved via `merge.context.sections['section name']`. Remember, however, that when several contexts need to be merged (for example, when the Print context is attached to an email), the script needs to check if the current context is of the type that contains the desired section (for example: `if (merge.context.type == ContextType.PRINT) {}`). When sections in different contexts have the same name, it is safer to use `merge.template.contexts.ContextType.sections["section name"]`.

Fields

Field	Type	Description
"background" on page 951	Background	**Print sections only.** Used to set a PDF background on a Print section. See "Control Script: Setting a Print section's background" on page 438 and "BackgroundResource" on page 962.
enabled	Boolean	Enables or disables this section for output (see "Examples" on page 947). Note that even if a Print section is disabled, the `part` and `restartPageNumber` fields are still effective to define the parts division and page numbering over multiple sections when applicable. The default enabled state for sections (before any control script runs) is as follows: • For **Email** channel requests on the **Print** context **all** Print sections are enabled by default. It is possible to enable different or multiple sections to control which sections will be attached to the email. • For **Email** channel requests, the requested section is enabled by default. It is possible to output another section by disabling the requested section and enabling another section. • For **Print** channel requests on the **Print** context **all** sections are enabled by default. It is possible to enable different or multiple sections to control which sections will be outputted.

Field	Type	Description
headers	String	***DEPRECATED. Use the addHeader() function instead.*** **Email sections only**. Used to set custom email headers.
height	String	**Print sections only.** The height of the section using a Measurement string. For example, "11in" is 11 inches. In a Control Script this property can be used to set the height of the section. In other scripts it is only possible to get the height of the section (read-only).
"html()" on page 958		Gets or sets the HTML of the body element.
margins	Margins	**Print sections only**. Used to set the print margins of a section. You can set the bottom, left, right and top margin using a Measurement string; for example, "2in" sets a margin to 2 inches.
minPages	Number	Minimum number of pages in the section. The default is 1.
name	String	Used to get or set the name of the section. Note that section names must be unique and that sections cannot have an integer as its name. The name should always include alphanumeric characters. To rename email attachments, use the field `part`.
ownerPassword	String	**Print sections only**. Used to set the owner password for a PDF attachment.* Setting only the owner password creates a secured

Field	Type	Description
		PDF that can be freely viewed, but cannot be manipulated unless the owner password is provided. (Note that the recipient needs Adobe Acrobat to do this, because the Acrobat Reader does not allow users to enter the owner password.) See "Control Script: Securing PDF attachments" on page 443.
pagination	"Pagination" on page 988	**Post Pagination scripts only.** Contains the total page count, sheet count and start/end page numbers of a single section.
part	String	Name for the part. `part` is used to specify where a new part starts and the title for the part. This is used to split Email attachments. The Email output can, for example, attach 3 PDFs generated from the Print context. The part name will be used as the file name for the attachment. See "Parts: splitting and renaming email attachments" on page 436.
password	String	**Print sections only**. Used to set the user password and owner password for a PDF attachment to the same value. See "Control Script: Securing PDF attachments" on page 443.*
restartPageNumber	Boolean	**Print sections only**. Enables or disables a restart of the page numbering. When generating Print output this can be used to let page numbering continue over multiple sections. The default value is `false`, meaning that each section will start with page 1 (to emulate behavior of previous versions).

Field	Type	Description
"sheetConfig" on page 954	SheetConfig	**Print sections only**. Overrides the Master Page, Media and Duplex printing options of first/middle/last/single or all sheets in a Print section.
width	String	**Print sections only.** The width of the section using a Measurement string. For example, "8.5in" is 8.5 inches. In a Control Script this property can be used to change the width of the section. In other scripts it is only possible to get the width of the section (read-only).

***** The password(s) should be set on the first Print section when producing a single attachment, or on the first section of each part when producing multiple attachments. Each of the parts (attachments) may have a different (or no) set of passwords.
Passwords set in the Control Script override the password set through the Email PDF password script (see "Email PDF password" on page 186). This allows you to change or remove the password from a specific part. Removal is done by setting the password field to null or "" (empty string).

Note

When a Control Script changes the size of a section, it should also change the size of the linked Media; this is not done automatically. While the output may still look good, a size mismatch can cause an issue if a script or another component assumes that the media size matches the section size.
In case of a size mismatch a preflight will show a warning (see "Doing a Preflight" on page 405).

Functions

> **Note**
>
> For cloned sections, functions are not available.

Function	Description
"clone()" on page 957	Clone this section. See "Dynamically adding sections (cloning)" on page 440.
addAfter()	Add a cloned section after this section.
addBefore()	Add a cloned section before this section.
addHeader (key, value)	**Email sections only**. Used to set custom email headers. The function expects a key and a value (both of type string). For examples, see "Adding custom ESP handling instructions" on page 1023.
"html()" on page 958	Sets the initial HTML of the body element.
"paginate()" on page 988	**Post Pagination Scripts only**. Paginates the content of a Print section.

Examples

Conditionally skipping or printing Print sections

This script disables all Print sections and then re-enables one of them, depending on a value in the current record.

```
var printSections = merge.template.contexts.PRINT.sections;
printSections['Section EN'].enabled = false;
printSections['Section FR'].enabled = false;

if(record.fields.Language === 'FR'){
       printSections['Section FR'].enabled = true;
```

```
} else {
        printSections['Section EN'].enabled = true;
}
```

Selecting different sections for Print output and Email PDF attachment

This script selects a different Print section for output, depending on the output channel (Email or Print).

```
var printSections = merge.template.contexts.PRINT.sections;

if(merge.channel === Channel.EMAIL){
        printSections['Section 1'].enabled = false;
        printSections['Section 2'].enabled = true;
}

if(merge.channel === Channel.PRINT){
        printSections['Section 1'].enabled = true;
        printSections['Section 2'].enabled = false;
}
```

Setting the name of Email PDF attachments

This script renames the file name of an attachment by setting the part name of a section (see "Parts: splitting and renaming email attachments" on page 436).

```
var section = merge.template.contexts.PRINT.sections['Section 1'];
section.part = 'Invoice ' + record.fields['InvoiceNo'];
```

Controlling multiple Email attachments

The following script attaches the following sections to an email:

- Print section 3 + 4 as attachment with continued page numbers
- Print section 6 as separate attachment (also see "Parts: splitting and renaming email attachments" on page 436)

```
if (channel == Channel.EMAIL) { // only when generating Email
output
if (merge.context.type == ContextType.PRINT) {
        merge.context.sections['Section 1'].enabled = false;
        merge.context.sections['Section 2'].enabled = false;
        merge.context.sections['Section 3'].enabled = true;
        merge.context.sections['Section 3'].part = "PDFAttach1";
        merge.context.sections['Section 4'].enabled = true;
```

```
        merge.context.sections['Section 4'].restartPageNumber = false;
        merge.context.sections['Section 5'].enabled = false;
        merge.context.sections['Section 6'].enabled = true;
        merge.context.sections['Section 6'].part = "PDFAttach2";
}
}
```

Note

For another example, see this how-to: Output sections conditionally.

Note

If the Email PDF Password Script Wizard defines a password, and a template has a Control Script that creates multiple PDF attachments, all the attachments are secured by the same password by default. Using a Control Script, you can set set different passwords for attachments; see "Control Script: Securing PDF attachments" on page 443.

Positioning the background of a Print section

These scripts both set the background of a Print section to the same PDF, but they position it differently.

Using abolute positioning

```
var activeSection = merge.template.contexts.PRINT.sections['Section
1'];
activeSection.background.source = BackgroundResource.RESOURCE_PDF;
activeSection.background.position = MediaPosition.ABSOLUTE;
activeSection.background.left = "10mm";
activeSection.background.top = "10mm";
activeSection.background.url = "images/somepage.pdf";
```

Scaling to Media size

```
var activeSection = merge.template.contexts.PRINT.sections['Section
1'];
activeSection.background.source = BackgroundResource.RESOURCE_PDF;
```

```
activeSection.background.position = MediaPosition.FIT_TO_MEDIA;
activeSection.background.url = "images/somepage.pdf";
```

See also: "BackgroundResource" on page 962, "MediaPosition" on page 966 and "Control Script: Setting a Print section's background" on page 438.

Cloning Print sections

For background information on cloning Print sections, see: "Dynamically adding sections (cloning)" on page 440.

Cloning a section based on the number of records in a detail table

This script creates as many clones of a section as there are records in a detail table. It assigns the new sections a unique name.

```
var printSections = merge.template.contexts.PRINT.sections;
var numClones = record.tables['detail'].length;
for( var i = 0; i < numClones; i++){
	var clone = printSections["Section 1"].clone();
	clone.name = "my_section_clone_" + i;
	printSections["Section 1"].addAfter(clone);
}
```

Cloning a section based on data and assign a background PDF

This script clones a section based on data fields. It disables the source section first and then calls the `addPolicy()` function. `addPolicy()` clones the section, renames it and sets a PDF from the resources as its background. It explicitly enables the clone and then adds it to the Print context.

```
var printSections = merge.template.contexts.PRINT.sections;
merge.template.contexts.PRINT.sections["Policy"].enabled = false;
if(record.fields.policy_a == 1) {
	addPolicy('a');
}
if(record.fields.policy_b == 1) {
	addPolicy('b');
}
function addPolicy(policy){
	var resourceUrl = 'images/policy-' + policy + '.pdf';
	var clone = printSections["Policy"].clone();
	clone.name = "policy_" + policy;
```

```
        clone.background.url = resourceUrl;
        clone.enabled = true;
        printSections["Policy"].addAfter(clone);
}
```

background

The `background` object holds the PDF background of a Print section (see "section" on page 978 and "Control Script: Setting a Print section's background" on page 438).

> **Note**
>
> Setting a page range using the `start` and `end` fields automatically sets `background.allPages` to `false`.
> When you first define a page range and then set `background.allPages` to `true`, the page range will be disabled.

Fields

Field	Type	Description
allPages	Boolean	Show all pages from the PDF.
end	Number	The end page of the PDF to use as a background for the section.
left	Measurement	The left offset of the PDF background (only when absolute positioning is selected).
position	"MediaPosition" on page 966	Set the position of the PDF background (Absolute, centered, fit to media).
rotation	Number	Set the rotation of the PDF background to 0, 90, 180 or -90 degrees.
scale	Number or Object	Set the size of the PDF background as a percentage of the original image. Accepted values are:

Field	Type	Description
		• A numeric value. • A string containing a number and percentage character: "<number>%". • An object with two properties, "x" and "y", with numeric values: "{ x: 100, y: 100 }". If x and y are different the image will be stretched.
source	"BackgroundResource" on page 962	Set the source of the PDF background (NONE, Datamapper, PDF Resource). **Note** Setting the background to NONE does not reset any other option, such as the position and scale of the background. **Note** **DataMapper** cannot be used in PrintShopMail Connect. The DataMapper is included only in PlanetPress Connect and PReS Connect.
start	Number	The start page of the PDF to use as a background for the section.
top	Measurement	The top offset of the PDF background (only when absolute positioning is selected).
url	String	The location of the PDF to use as a background for the section. Three different forms of URLs are supported: • A relative path to an image resource; this

Field	Type	Description
		always starts with "images/". For a file named background.pdf, stored inside the template resources, the URL would be `images/background.pdf`. • A URL (either `file:`, `http:`, or `https:`). The complete syntax for an external file is: file://<host>/<path>. If the host is "localhost", it can be omitted, resulting in file:///<path>, for example: `file:///c:/resources/images/image.jpg`. • An absolute Windows-style path. A relative path and a URL are expected to be URL encoded. For example, a space needs to be encoded as `%20`. An absolute Windows-style path does not need to be URL encoded, but in JavaScript any backslashes in a String need to be double-escaped: "C:\\Images\\Image.pdf".

Example

This scripts sets a background on a Print section using absolute positioning.

```
var activeSection = merge.template.contexts.PRINT.sections['Section
1'];
activeSection.background.source = BackgroundResource.RESOURCE_PDF;
activeSection.background.url = "images/somepage.pdf";
activeSection.background.position = MediaPosition.ABSOLUTE;
activeSection.background.left = "10mm";
activeSection.background.top = "10mm";
```

You could replace the last three lines of the previous script by the following line to scale the Print section background to Media size:

```
activeSection.background.position = MediaPosition.FIT_TO_MEDIA;
```

Tip

For more examples, see "Control Script: Setting a Print section's background" on page 438.

sheetConfig

Sheet configuration settings are the Master Page, Media and Duplex printing options of first/middle/last/single or all sheets in a **Print** section. The `sheetConfig` object holds these options and can be used to set (or rather, override) them via a Control Script (see "Control Scripts" on page 429 and "Control Script API" on page 931). This is especially useful when you need identical sections with different sheet configuration settings.

The `sheetConfig` object can be retrieved via the `section` object (see "section" on page 978); see the example below.

Fields

The fields of the `sheetConfig` object correspond to settings in the Sheet Configuration dialog; see "Sheet Configuration dialog" on page 546.

Field	Type	Description
duplex	Boolean	Enables or disables duplex (two-sided) printing.
facingPages	Boolean	(Only with duplex printing.) When `facingPages` is set to `true`, the margins of the section switch alternately, so that pages are printed as if in a magazine or book.
mediaRotation	Number	Rotates the Media (to be more accurate: the Virtual Stationery images specified for this Media) by 0, 90, 180, or -90 degrees.
omitEmptyBackside	Boolean	(Only with duplex printing.) Resets a page to Simplex if it has an empty back side. This may reduce volume printing costs.

Field	Type	Description

positions	Positions (see "Position" below)	Used to modify the Master Page and Media and to allow content on sheets in different positions; for example:
		`var section1 = merge.template.contexts.PRINT.sections[0];`
		`section1.sheetConfig.positions.all.media = "Media 1";`
		For an overview of the available settings in each position see "Position" below.
		Can also be used to repeat the sheet configuration *within* a document.

tumble	Boolean	(Only with duplex printing.) When `tumble` is set to `true`, pages are printed like in a calendar. (On Portrait output, this would be equivalent to short-edge duplex.)

Example

This script retrieves a section and changes its sheet configuration settings.

```
let section = merge.template.contexts.PRINT.sections["Section 1"];
section.sheetConfig.duplex = true;
section.sheetConfig.omitEmptyBackside = true;
section.sheetConfig.facingPages = true;
section.sheetConfig.mediaRotation = 0;
```

Position

The `positions` property of the `sheetConfig` object (see "sheetConfig" on the previous page) can be used to repeat the sheet configuration after a certain number of pages, or to retrieve and modify settings for sheets in different positions. Available positions are:

- **single**: The only sheet in a section.
- **first**: The first sheet in the section.
- **middle**: All after the first sheet and before the last sheet in a section.

- **last**: The last sheet in a section.
- **all**: All sheets in the section.

> **Note**
>
> Settings for all sheets may conflict with settings for a particular position. In such cases, the setting that was specified last overrides the other.

Each of these positions has the following fields.

Field	Type	Description
allowContent	AllowContent	Used to specify on which sides of a sheet (in a certain position) content is allowed. Possible values: • AllowContent.ALL_SIDES • AllowContent.FRONT_ONLY (only with duplex printing) • AllowContent.BACK_ONLY (only with duplex printing) • AllowContent.NONE (not allowed on all, single and middle positions)
masterBack	String	Specifies the Master Page to use on the backside of sheets in this position.
masterFront	String	Specifies the Master Page to use on the front of sheets in this position.
media	String	Specifies the Media to use with sheets in this position.
repeat	Number	Repeat the sheet configuration every *n* number of pages. This is useful e.g. when documents can be so long that they cannot fit in one envelope.

Note

To remove a Media or Master Page via script, give it the value `undefined`, or `null`, or an empty string (""); see the examples below.

Examples

This script retrieves a Print section and modifies a number of settings for all of its sheets.

```
let section = merge.template.contexts.PRINT.sections["Section 1"];
section.sheetConfig.positions.all.allowContent = AllowContent.ALL_
SIDES;
section.sheetConfig.positions.all.media = "MyMedia";
section.sheetConfig.positions.all.masterFront = "Master page 1";
section.sheetConfig.positions.all.masterBack = undefined; // or
null, or an empty string
```

The following script ensures that empty backsides of single sheets are omitted.

```
let section = merge.template.contexts.PRINT.sections["Section 1"];
section.sheetConfig.positions.single.omitMasterOnEmptyBackside =
true;
```

clone()

This function returns a copy of one HTML element or of a set of HTML elements, which can be:

- The elements that match the selector of a script (see "results" on page 972).
- One element that matches the selector of a script that runs for "Each matched element" (see "this" on page 891 and "Setting the scope of a script" on page 399).
- The elements returned by a query in the template (see "query()" on page 845).

See also: "Dynamically adding sections (cloning)" on page 440.

To duplicate an existing template element, clone it before calling `append()`; see "append()" on page 900.

Examples

This script performs an iteration over the elements in the `results` (the elements that match the selector of the script).

```
var row = query("tbody tr", results).clone();
query("tbody", results).append(row);
```

The following script clones an existing table row to match the number of rows in a detail table. Afterwards it iterates over the rows to populate the fields.

```
// Create the number of rows based on the records in the detail
table
// We start at 1 so the boilerplate row is used too and there is no
need to delete that row
for(var r = 1; r < record.tables['detail'].length; r++) {
results.parent().append(results.clone());
}

// Iterate over the rows and populate them with the data from the
accompanying data row
query("#table_2 > tbody > tr").each(function(i) {
this.find('@ItemNumber@').text( record.tables['detail'][i].fields
["ItemNumber"]);
this.find('@ItemOrdered@').text( record.tables['detail'][i].fields
["ItemOrdered"]);
this.find('@ItemTotal@').text( record.tables['detail'][i].fields
["ItemTotal"]);
this.find('@ItemDesc@').text( record.tables['detail'][i].fields
["ItemDesc"]);
this.find('@nr@').text(i);
});
```

The following script clones and populates a boilerplate row. Once completed you will need to hide the boilerplate row.

html()

In a Control Script, the `html()` function of a section or Master Page can be used to get the initial contents of its <body>, and modify them. This makes it possible, for example, to populate a section or Master Page with elements retrieved from a Content Management system, before Standard Scripts run.

html()

Gets the initial contents of the <body> of a section or Master Page.

html(value)

Replaces the contents of the <body> of a section or Master Page with the supplied value. This function is only available in Control Scripts. See also: "section" on page 978 and "masterpage" on page 934.

value

A String that may contain HTML tags.

Examples

The following script uses `html()` to retrieve the contents of a section and add a paragraph to it.

```
var foo = merge.context.sections["Section 1"].html();
foo += "<p>hello world</p>";
merge.context.sections["Section 1"].html( foo );
```

The following script loads a snippet based on the value of a field, and then replaces the content of a Print section with the snippet using `html()`.

```
var mySection = merge.context.sections["Section 1"];
var promoTxt = loadhtml('snippets/promo-' + record.fields['City'] +
'.html');
mySection.html(promoTxt);
```

paginate()

This method of the `section` object (see "section" on page 978) triggers pagination of the current section. The pagination process re-establishes page boundaries, updates page numbers and page counts, and reapplies Master Pages.
Depending on whether page numbering restarts in each section this may affect the page numbers in other sections as well (see "Configuring page numbers" on page 152).
When the pagination process has ended, the script resumes.

The 'current section' is always a Print section, since this method can only be used in Post Pagination Scripts, and Post Pagination Scripts only run on the Print context (see "Post Pagination Scripts" on page 445).

You only need to call `merge.section.paginate()` in a Post Pagination Script if the script has added or removed content to such an extent that the page boundaries need to be renewed.

For an example see: "Creating a Table Of Contents" on page 447.

template

The `template` object represents the template with all its contexts and sections. It is used frequently in Control Scripts (see "Control Scripts" on page 429 and "Control Script API" on page 931) but it can also be used in Standard Scripts.

It is retrieved via the merge object: `merge.template` (see "merge" on page 989).

Which contexts are available in the template can be queried using `merge.template.contexts`. To get access to a specific context, you have to specify the ContextType (see "ContextType" on page 965).

Field	Type	Description
contexts	Array	Array of contexts (see "context" on page 969) available in the template. The contexts contain the sections (see "section" on page 978).
images	Array	The list of image resources (see "ImageResource" on page 965) included in the template.
masterpages	Array	Array of Master Pages (see "masterpage" on page 934) available in the template.
"media" on page 937	Array	Media available to this template (see "Media" on page 159). For each of them you can specify, enable and position the stationery's front and back.
"properties" on the next page	Properties	This object contains all default properties of the template as well as any custom

Field	Type	Description
		properties. (On the menu, select **File > Properties** to view and complement the file properties. See "File Properties dialog" on page 496).

Example

The following Control Script retrieves two Print sections. Then, depending on a value in the current record, it enables one section or the other, so that only one of the two sections appears in the output.

```
var printSections = merge.template.contexts.PRINT.sections;
printSections['Section EN'].enabled = false;
printSections['Section FR'].enabled = false;

if(record.fields.Language === 'FR'){
	printSections['Section FR'].enabled = true;
} else {
	printSections['Section EN'].enabled = true;
}
```

properties

The `properties` object inside the `template` object (see "template" on the previous page) contains all default properties of the template file as well as any custom properties.

To view and complement the file properties, select **File > Properties** on the menu. See "File Properties dialog" on page 496.

Following are the default properties.

Field	Type	Description
application	String	Application version when the document was last saved

Field	Type	Description
author	String	Original author of the document
company	String	company name
created	Date	Date and time on which the document was created
customProperties	List	A list of defined custom properties
description	String	Description of the document
file	String	File name of the document
keywords	String	Semicolon-separated list of keywords
modified	Date	Date and time on which the document was last saved

Example

This script stores a default property (author) and a custom property in variables.

```
var author = merge.template.properties.author;
var myProperty = merge.template.customProperties.myPropertyName;
```

BackgroundResource

BackgroundResource is an enumeration for the types of background resources for a Print section (see "background" on page 951 and "section" on page 978).

A Print section can be retrieved in script using `merge.template.contexts.ContextType.sections["section name"]`, for example `merge.template.contexts.PRINT.sections["Section EN"]`.

Field	Description
DATAMAPPER_PDF	A PDF file retrieved via the active data mapping configuration. This can be the PDF file that was used as input file, or another type of input file, converted to PDF. **Note** **From DataMapper input** cannot be used in PrintShopMail Connect. The DataMapper is included only in PlanetPress Connect and PReS Connect.
NONE	No PDF background.
RESOURCE_PDF	A PDF or other image file which is stored in the template or on the network. Note that it isn't possible to use a remotely stored image file as a section's background.

Example

The following script sets the background for a section called 'Policy' to `RESOURCE_PDF` and specifies a path for it, using a data value:

```
// Enable the section background and specify that the PDF should be
read
// from a resource file
merge.template.contexts.PRINT.sections['Policy'].background.source
= BackgroundResource.RESOURCE_PDF;

// Specify the path
var resourceUrl = 'images/policy-' + record.fields.policy + '.pdf';
merge.template.contexts.PRINT.sections['Policy'].background.url =
resourceUrl;
```

Note

To learn how to set a PDF file as a background image on a Print section without a Control Script, see "Using a PDF file or other image as background" on page 142.

Channel

`Channel` is an enumeration for the output channels. The active output channel is registered in `merge.channel`.

The channel doesn't change when the output consists of different contexts. When generating email, for example, the channel is EMAIL, even when merging the Print context to attach it to the email.

Value	Description
EMAIL	The merge request is for output to Email.
PRINT	The merge request is for output to Print.
THUMBNAIL	The merge request is for generating a template preview.

Example

The following Control Script selects different sections for Print output and for Email with the Print context attached to it.

```
var printSections = merge.template.contexts.PRINT.sections;

if(merge.channel === Channel.EMAIL){
	printSections['Section 1'].enabled = false;
	printSections['Section 2'].enabled = true;
}

if(merge.channel === Channel.PRINT){
	printSections['Section 1'].enabled = true;
	printSections['Section 2'].enabled = false;
}
```

ContextType

ContextType is an enumeration for the context types.

The type of the context that is going to be merged next can be retrieved via `merge.context.type`.

The context type needs to be specified when retrieving a section with `merge.template.contexts.ContextType.sections["section name"]`, for example `merge.template.contexts.PRINT.sections["Section EN"]`.

Value	Description
HTML_EMAIL	The context is the Email context.
PRINT	The context is the Print context.

Example

This script retrieves two Print sections. Then, depending on a value in the current record, it enables one section or the other, so that only one of the two sections appears in the output.

```
var printSections = merge.template.contexts.PRINT.sections;
printSections['Section EN'].enabled = false;
printSections['Section FR'].enabled = false;

if(record.fields.Language === 'FR'){
	printSections['Section FR'].enabled = true;
} else {
	printSections['Section EN'].enabled = true;
}
```

ImageResource

The `ImageResource` object represents an image resource in a template.

In script, an `ImageResource` is retrieved via the `images` Array which contains a list of all image resources that are included in the current template (see "template" on page 960).
For example: `var myImage = merge.template.images[0];`.

Functions and fields

Field	Type	Description
modified	Number	Timestamp of the last modification of the image resource, for example: 1566660504000.
name	String	The base name of the image resource, for example: "cat.jpg".
path	String	The path of the image resource, relative to the template, for example: "images/cat.jpg".

Example

The following script adds an image called 'dog.jpg' to the content of the template if the image can be found in the template's resources:

```
const image = merge.template.images.find(image => image.name == 'dog.jpg');
if (image) {
 results.after(`<img src=${image.path}>`);
}
```

MediaPosition

In a Control Script, the `position` is an enumeration for the position of background resources for a Print section. It is retrieved and set via `background.position`.

Field	Description
ABSOLUTE	Places the PDF at a specific location on the page. Set the background's top (`background.top`) and left (`background.left`) measured from the top and left side of the section.
CENTERED	Centers the PDF on the page, vertically and horizontally.
FIT_TO_ MEDIA	Stretches the PDF to fit the page size.

Examples

This script applies **absolute positioning** to the background of a Print section.

```
var activeSection = merge.template.contexts.PRINT.sections['Section
1'];
activeSection.background.source = BackgroundResource.RESOURCE_PDF;
activeSection.background.position = MediaPosition.ABSOLUTE;
activeSection.background.left = "10mm";
activeSection.background.top = "10mm";
activeSection.background.url = "images/somepage.pdf";
```

The next script scales the background of a Print section to the size of the **Media**.

```
var activeSection = merge.template.contexts.PRINT.sections['Section
1'];
activeSection.background.source = BackgroundResource.RESOURCE_PDF;
activeSection.background.position = MediaPosition.FIT_TO_MEDIA;
activeSection.background.url = "images/somepage.pdf";
```

Post Pagination Script API

The table below lists the functions and objects that are **only** available in Post Pagination Scripts. Click through to the object to find a description and sample scripts.
See "Post Pagination Scripts" on page 445 for information about this kind of scripts, how to insert them and what you can do with them.

> **Note**
>
> Post Pagination Scripts only apply to the Print context. These functions and objects cannot be used with other contexts.

> **Tip**
>
> It is possible to *define* a function in a Control Script which calls these functions, as long as that function is only invoked from a Post Pagination Script.

Object or function	Usage
merge.context."query (selector)" on page 971	Call `merge.context.query(selector)` to run a query across all sections in the Print context.
merge.section."paginate()" on page 988	A call to `merge.section.paginate()` triggers pagination of the current section. Note: `section` is assumed to refer to a variable that holds a Section object (see "Retrieving a section" on page 979).
merge.section.pagination	Used to get the total page count, sheet count and start/end page numbers for a single section. See "Pagination" on page 988. Note: `section` is assumed to refer to a variable that holds a Section object (see "Retrieving a section" on page 979).
merge.pagination	Used to get the total page count, sheet count and start/end page numbers for all sections in the Print context after pagination. See "Pagination" on page 991.
"PaginationInfo" on page 971	This object contains essential information about one element in order to create a table of contents (see "Creating a Table Of Contents" on page 447).
results."info()" on page 977	A call to results.info() returns pagination information (see "PaginationInfo" on page 971) for the first element in the result set of a query (see "query(selector)" on page 971).

Other objects that are available to Post Pagination Scripts

The list above isn't exhaustive: the objects listed in the Designer API (see "Standard Script API" on page 810) and in the Control Script API (see "Control Script API" on page 931) are also available in Post Pagination Scripts.

contentitem

The `contentitem` object holds the Print Content Item that will be written to the Connect database when generating Print output.

Its `properties` field allows to add custom properties to the Print Content Item, in the form of key-value pairs (a JSON string).

Custom properties can be utilized for further processing in a Workflow configuration with the Retrieve Items task. The Retrieve Items task retrieves custom properties along with the base record information (see Retrieve Items in Workflow's Online Help).

The `contentitem` object can be used in any type of script. However, if you want to add production information - such as the page, size, and position of elements after pagination - to the `properties`, you have to write a Post Pagination Script (see "Post Pagination Scripts" on page 445).

Field	Type	Description
properties	Properties	Key-value pairs containing custom properties. Note that a property value is always stored as a string in the database.

Example

This following code would add a property called 'myProperty' with the value 'myvalue':

```
contentitem.properties.myProperty = 'myvalue';
```

You can replace 'myProperty' and 'myvalue' with whatever name and value best suits the use case.

If the name of the property contains a space you'll need to put it between brackets and quotation marks:

```
contentitem.properties['name with spaces'] = 'value';
```

If the value is a number you don't need to put it between quotation marks:

```
contentitem.properties.custom_property = 123;
```

context

In a Control Script, the `context` object represents one context in the template.

Which contexts are available in the template can be queried using `merge.template.contexts`.

The context being merged can be queried using `merge.context`.

Field	Type	Description
sections	Array	Array of sections (see "section" on page 978) inside a particular context defined in the template. **Note**: When using `merge.context.sections` keep in mind that for example 'Section X' might only exist in your Print context, so using `merge.context.sections['Section X']` without enclosing it in the `if` statement `if (merge.context.type == ContextType.PRINT) {}` will yield an error when the script runs for other contexts. Alternatively, use the `template` object to access a specific context: `merge.template.contexts.PRINT.sections['Section X']`.
type	"ContextType" on page 965	The context type: PRINT, EMAIL or WEB.

Function	Description
"query (selector)" on the next page	Runs a query across all sections in the Print context. This function is only available in Post Pagination Scripts, which are only applied to the Print context. See "Post Pagination Scripts" on page 445.

Example

This script checks if the output channel is EMAIL and if the context to be merged is the Print context (which happens if the Print context is attached to an email). If this is the case, it includes and excludes certain Print sections from the output.

```
if (channel == Channel.EMAIL) {
        if (merge.context.type == ContextType.PRINT) {
                merge.context.sections['Section 1'].enabled = false;
                merge.context.sections['Section 2'].enabled = false;
                merge.context.sections['Section 3'].enabled = true;
```

```
        }
}
```

query(selector)

This function of the `context` object (see "context" on page 969) returns a result set, containing the HTML elements in **all** sections of the Print context that match the supplied CSS selector. The new result set is of the type `QueryResults`, just like the `results` object which is also the result of a (hidden) query (see "results" on the facing page), but it is **read-only**.

This function can only be called in a Post Pagination Script (see "Post Pagination Scripts" on page 445). It is indispensable in a script that creates a table of contents, as described in the following topic: "Creating a Table Of Contents" on page 447.

selector

A String containing a CSS selector. See https://www.w3schools.com/cssref/css_selectors.asp for CSS selectors and combinations of CSS selectors.

Example

This script returns all level 1 headings in all sections of the Print context.

```
var headings = merge.context.query("h1");
```

PaginationInfo

The PaginationInfo object is returned by the `info()` function of the `results` object (see "info()" on page 977 and "results" on the facing page). It contains information about where an element is located: on which page and which sheet, and the total number of pages and sheets in the page bundle in which the element is located.
A **page bundle** is a group of sections for which the Restart page numbering option is turned off. In other words, page numbering continues from the start of the first section to the end of the last section in the group. (See "Configuring page numbers" on page 152 and "Control Script: Page numbering" on page 433.)

The PaginationInfo object (as well as the `info()` function) is only available in Post Pagination Scripts (see "Post Pagination Scripts" on page 445). It is essential in a Post Pagination Script that creates a table of contents, as described in the following topic: "Creating a Table Of Contents" on page 447.

For the total page count, sheet count and start/end page numbers of a **single section**, use the `section` object's "Pagination" on page 988 (see also: "section" on page 978).

For the total page and sheet count of **all Print sections** together use the `merge` object's "Pagination" on page 991 (see also: "merge" on page 989).

Field	Type	Description
pageCount	Number	The number of pages in the page bundle associated with the section in which the first element of the results is located.
pageNo	Number	The page number on which the first element of the `results` is located.
"section" on page 978	Section	The section in which the element is located. This is always a Print section, since Post Pagination Scripts only operate on the Print context.
sheetCount	Number	The number of sheets in the page bundle associated with the section in which the first element of the `results` is located.
sheetNo	Number	The sheet number on which the first element of the `results` is located.

results

The `results` object (type: `QueryResults`) is the result of the query for HTML elements that match the selector of the script. The selector of a script can be specified in the Script Editor and is visible in the second column of the Scripts pane, next to the name of the script.

If, for example, a script would have the selector `p.onlyCanada`, the script would apply to all paragraphs that have the class `onlyCanada`. (Classes can be defined in the **Attributes** pane at the right: select the element in the content and type the class(es) in the **Class** field.)
The script could then use the `results` object to hide or show those paragraphs, depending on the value of the data field `Country` in the current record:

```
if (record.fields["Country"] == "CANADA") {
results.show();
```

```
} else {
results.hide();
}
```

Note

This object can't be used in Control Scripts, because they don't have a selector.

Tip

The easiest way to access the elements in a result set one by one, is by setting the scope of the script to "Each matched element" and using the this object (see "this" on page 891).

Property

Field	Type	Description
length	Number	Number of elements in this result set. Equivalent to calling size ().

Functions

The functions below can be called by the `results` object and by any other result set that is returned by a query, see "query()" on page 845.

Function	Description
"add()" on page 896	Adds elements to a set of HTML elements.
"addClass()" on page 897	Adds the specified class to each element in a set of HTML elements. Has no effect if the class is already present.
"after()" on	Inserts content after each element in a set of HTML elements..

Function	Description
page 898	
"append()" on page 900	Inserts content at the end of each element in a set of HTML elements.
"attr()" on page 903	Change the given attribute of the element or set of HTML elements with the given value.
"before()" on page 904	Inserts content before an element or before each element in a set of HTML elements.
"children()" on page 907	Returns the immediate children of an HTML element.
"clone()" on page 957	Returns a new result set containing a copy of each element in a set of HTML elements.
"closest()" on page 908	For each element in a set, this function gets the first parent element that matches a selector, by testing the element itself and traversing up through its ancestors in the DOM tree.
"css()" on page 909	Gets the value of a style property for the first element in set of HTML elements or sets one or more CSS properties for every element in a set of HTML elements.
"empty()" on page 912	Removes the contents (child elements and inner HTML) from one element or a set of elements in the template.
"filter()" on page 872	Returns a subset of the current result set.
"find()" on page 912	Performs a search for a text in the children of each element in a set of HTML elements, and returns a new result set with elements that surround the occurrences.
"get(index)" on	Returns the element (type: QueryResult) found at the supplied index in

Function	Description
page 873	a set of HTML elements.
"hasClass()" on page 913	Returns `true` if the first element in this result set has the specified class.
"height()" on page 913	Gets or sets the outer height of an element, including padding and borders.
"hide()" on page 914	Hides the HTML element or set of HTML elements.
"html()" on page 915	Replaces the inner HTML of the element or of each element in a set of HTML elements with the supplied value, or returns the HTML of the first element if no value is supplied.
"info()" on page 977	**Post Pagination Scripts only**. Returns pagination information for the first element in this result set.
is(selector)	Returns true if at least one of the elements in a set of HTML elements matches the supplied CSS selector.
"next()" on page 916	Returns the next sibling of each HTML element in the result set.
"overflows()" on page 977	Returns a boolean value indicating whether an HTML element overflows its box boundaries.
"pageref()" on page 919	Returns a **marker** that will be replaced with the element's page number after pagination. This only works for elements in the section that is currently being merged.
"parent()" on page 920	Returns the parents of the elements in a set of HTML elements.
position()	Returns information about the position (see NodePosition) of the first element in this result set, excluding margins. To include margins, call

Function	Description
	`position(true)`.
"prepend()" on page 921	Inserts content at the beginning of an HTML element or of each element in a set of HTML elements.
"prev()" on page 923	Returns the previous sibling of each HTML element in the result set.
"remove()" on page 925	Removes an HTML element or a set of HTML elements from the document.
"removeAttr()" on page 926	Removes the specified attribute from each element in this result set.
"removeClass()" on page 926	Removes the specified class from an element or from each element in a set of HTML elements. Has no effect if the class is not present.
"replaceWith()" on page 927	Replaces an HTML element or a set of HTML elements (with a snippet, for example). Returns the result set.
"show()" on page 928	Shows the HTML element or a set of HTML elements.
size()	Gets the number of elements in this result set. Equivalent to the `length` property.
"tagName()" on page 928	Returns the HTML tag name of the first element in this result set, in uppercase. For an example see: "Creating a Table Of Contents" on page 447.
"text()" on page 929	Replaces the text content of an HTML element or of each element in a set of HTML elements with the supplied value, or returns the text content of the first element if no value is supplied.
"width()" on	Gets or sets the outer width of an element, including padding and

Function	Description
page 929	borders.

info()

Returns pagination information for one HTML element or for the first element in a result set (see "results" on page 972) of a query across all sections in a Print context (see "query(selector)" on page 971).

The returned information is of the type `PaginationInfo` (see "PaginationInfo" on page 971).

This function can only be used in a Post Pagination Script ; see "Post Pagination Script API" on page 967.

For an example see: "Creating a Table Of Contents" on page 447.

overflows()

The `overflows()` method returns a boolean value indicating whether an HTML element overflows its box boundaries.

This function could for instance be used in a design where content needs to flow separately from the main text flow, or where a new, full-page, multi-column box should be inserted when the current one overflows.

Note that when called in a Standard script, this method runs *before* the pagination of Print output. After pagination - in the output, and in Preview mode - the element's overflow may have changed, especially when the element is combined with floating elements (i.e. elements that have the CSS style float) and located near a page-break.
To prevent this from happening, it may help to set the element's CSS style to `overflow:hidden`.
The `overflows()` method can also be used in a Post Pagination script.

> **Tip**
>
> You don't need a script to resize text in order to make it fit in a box. The Copy Fit feature automatically scales text to the available space (see "Copy Fit" on page 311).

Examples

```
var $box = query("#mybox");
while ( ! $box.overflows()) {
        //do something
}
```

A slightly shorter version:

```
while ( ! query("#mybox").overflows()) {
        //do something
}
```

The following script selects three boxes that all have the class box (selector: .box). It loops over them, inserting as many products from an array as possible into the box. In fact, it inserts one too many and removes the last one before moving on to the next box.

```
var products =
["Appetizer","Beans","Beef","Lettuce","Sprouts","Coconut","Juice","
Soup","Coriander","Cheese","Pasta","Sugar","Vinegar","Bread"];
var i = 0;
results.each( function() {
        // Add elements to the box until it overflows.
        while ( ! this.overflows() && i < products.length ) {
                this.append("<p>" + products[i] + "</p>");
                i++;
        }
        // Go one step back unless we processed all items in our array.
        if( i < products.length ) {
                query("p:last-child", this).remove();
                i--;
        }
});
```

section

The section object can be used to query and modify how the section (and the related context) will be outputted. It is one of the most important objects in Control Scripts (see "Control Scripts" on page 429 and "Control Script API" on page 931).

Retrieving a section

A section can be retrieved using `merge.template.contexts.ContextType.sections["section name"]`, for example: `merge.template.contexts.PRINT.sections["Section EN"]`.

A section can also be retrieved via `merge.context.sections['section name']`. Remember, however, that when several contexts need to be merged (for example, when the Print context is attached to an email), the script needs to check if the current context is of the type that contains the desired section (for example: `if (merge.context.type == ContextType.PRINT) {}`). When sections in different contexts have the same name, it is safer to use `merge.template.contexts.ContextType.sections["section name"]`.

Fields

Field	Type	Description
"background" on page 951	Background	**Print sections only.** Used to set a PDF background on a Print section. See "Control Script: Setting a Print section's background" on page 438 and "BackgroundResource" on page 962.
enabled	Boolean	Enables or disables this section for output (see "Examples" on page 984). Note that even if a Print section is disabled, the `part` and `restartPageNumber` fields are still effective to define the parts division and page numbering over multiple sections when applicable. The default enabled state for sections (before any control script runs) is as follows: • For **Email** channel requests on the **Print** context **all** Print sections are enabled by default. It is possible to enable different or multiple sections to control which sections will be attached to the email.

Field	Type	Description
		• For **Email** channel requests, the requested section is enabled by default. It is possible to output another section by disabling the requested section and enabling another section. • For **Print** channel requests on the **Print** context **all** sections are enabled by default. It is possible to enable different or multiple sections to control which sections will be outputted.
headers	String	***DEPRECATED. Use the addHeader() function instead.*** **Email sections only**. Used to set custom email headers.
height	String	**Print sections only.** The height of the section using a Measurement string. For example, "11in" is 11 inches. In a Control Script this property can be used to set the height of the section. In other scripts it is only possible to get the height of the section (read-only).
"html()" on page 958		Gets or sets the HTML of the body element.
margins	Margins	**Print sections only**. Used to set the print margins of a section. You can set the bottom, left, right and top margin using a Measurement string; for example, "2in" sets a margin to 2 inches.
minPages	Number	Minimum number of pages in the section. The default is 1.

Field	Type	Description
name	String	Used to get or set the name of the section. Note that section names must be unique and that sections cannot have an integer as its name. The name should always include alphanumeric characters. To rename email attachments, use the field `part`.
ownerPassword	String	**Print sections only**. Used to set the owner password for a PDF attachment.* Setting only the owner password creates a secured PDF that can be freely viewed, but cannot be manipulated unless the owner password is provided. (Note that the recipient needs Adobe Acrobat to do this, because the Acrobat Reader does not allow users to enter the owner password.) See "Control Script: Securing PDF attachments" on page 443.
pagination	"Pagination" on page 988	**Post Pagination scripts only.** Contains the total page count, sheet count and start/end page numbers of a single section.
part	String	Name for the part. `part` is used to specify where a new part starts and the title for the part. This is used to split Email attachments. The Email output can, for example, attach 3 PDFs generated from the Print context. The part name will be used as the file name for the attachment. See "Parts: splitting and renaming email attachments" on page 436.
password	String	**Print sections only**. Used to set the user password and owner password for a PDF attachment to the same value. See "Control

Field	Type	Description
		Script: Securing PDF attachments" on page 443.*
restartPageNumber	Boolean	**Print sections only**. Enables or disables a restart of the page numbering. When generating Print output this can be used to let page numbering continue over multiple sections. The default value is `false`, meaning that each section will start with page 1 (to emulate behavior of previous versions).
"sheetConfig" on page 954	SheetConfig	**Print sections only**. Overrides the Master Page, Media and Duplex printing options of first/middle/last/single or all sheets in a Print section.
width	String	**Print sections only.** The width of the section using a Measurement string. For example, "8.5in" is 8.5 inches. In a Control Script this property can be used to change the width of the section. In other scripts it is only possible to get the width of the section (read-only).

* The password(s) should be set on the first Print section when producing a single attachment, or on the first section of each part when producing multiple attachments. Each of the parts (attachments) may have a different (or no) set of passwords.
Passwords set in the Control Script override the password set through the Email PDF password script (see "Email PDF password" on page 186). This allows you to change or remove the password from a specific part. Removal is done by setting the password field to null or "" (empty string).

Note

When a Control Script changes the size of a section, it should also change the size of the linked Media; this is not done automatically. While the output may still look good, a size mismatch can cause an issue if a script or another component assumes that the media size matches the section size.
In case of a size mismatch a preflight will show a warning (see "Doing a Preflight" on page 405).

Functions

Note

For cloned sections, functions are not available.

Function	Description
"clone()" on page 957	Clone this section. See "Dynamically adding sections (cloning)" on page 440.
addAfter()	Add a cloned section after this section.
addBefore()	Add a cloned section before this section.
addHeader (key, value)	**Email sections only**. Used to set custom email headers. The function expects a key and a value (both of type string). For examples, see "Adding custom ESP handling instructions" on page 1023.
"html()" on page 958	Sets the initial HTML of the body element.
"paginate()" on page 988	**Post Pagination Scripts only**. Paginates the content of a Print section.

Examples

Conditionally skipping or printing Print sections

This script disables all Print sections and then re-enables one of them, depending on a value in the current record.

```
var printSections = merge.template.contexts.PRINT.sections;
printSections['Section EN'].enabled = false;
printSections['Section FR'].enabled = false;

if(record.fields.Language === 'FR'){
	printSections['Section FR'].enabled = true;
} else {
	printSections['Section EN'].enabled = true;
}
```

Selecting different sections for Print output and Email PDF attachment

This script selects a different Print section for output, depending on the output channel (Email or Print).

```
var printSections = merge.template.contexts.PRINT.sections;

if(merge.channel === Channel.EMAIL){
	printSections['Section 1'].enabled = false;
	printSections['Section 2'].enabled = true;
}

if(merge.channel === Channel.PRINT){
	printSections['Section 1'].enabled = true;
	printSections['Section 2'].enabled = false;
}
```

Setting the name of Email PDF attachments

This script renames the file name of an attachment by setting the part name of a section (see "Parts: splitting and renaming email attachments" on page 436).

```
var section = merge.template.contexts.PRINT.sections['Section 1'];
section.part = 'Invoice ' + record.fields['InvoiceNo'];
```

Controlling multiple Email attachments

The following script attaches the following sections to an email:

- Print section 3 + 4 as attachment with continued page numbers
- Print section 6 as separate attachment (also see "Parts: splitting and renaming email attachments" on page 436)

```
if (channel == Channel.EMAIL) { // only when generating Email
output
if (merge.context.type == ContextType.PRINT) {
        merge.context.sections['Section 1'].enabled = false;
        merge.context.sections['Section 2'].enabled = false;
        merge.context.sections['Section 3'].enabled = true;
        merge.context.sections['Section 3'].part = "PDFAttach1";
        merge.context.sections['Section 4'].enabled = true;
        merge.context.sections['Section 4'].restartPageNumber = false;
        merge.context.sections['Section 5'].enabled = false;
        merge.context.sections['Section 6'].enabled = true;
        merge.context.sections['Section 6'].part = "PDFAttach2";
}
}
```

Note

For another example, see this how-to: Output sections conditionally.

Note

If the Email PDF Password Script Wizard defines a password, and a template has a Control Script that creates multiple PDF attachments, all the attachments are secured by the same password by default. Using a Control Script, you can set set different passwords for attachments; see "Control Script: Securing PDF attachments" on page 443.

Positioning the background of a Print section

These scripts both set the background of a Print section to the same PDF, but they position it differently.

Using abolute positioning

```
var activeSection = merge.template.contexts.PRINT.sections['Section
1'];
activeSection.background.source = BackgroundResource.RESOURCE_PDF;
activeSection.background.position = MediaPosition.ABSOLUTE;
activeSection.background.left = "10mm";
activeSection.background.top = "10mm";
activeSection.background.url = "images/somepage.pdf";
```

Scaling to Media size

```
var activeSection = merge.template.contexts.PRINT.sections['Section
1'];
activeSection.background.source = BackgroundResource.RESOURCE_PDF;
activeSection.background.position = MediaPosition.FIT_TO_MEDIA;
activeSection.background.url = "images/somepage.pdf";
```

See also: "BackgroundResource" on page 962, "MediaPosition" on page 966 and "Control Script: Setting a Print section's background" on page 438.

Cloning Print sections

For background information on cloning Print sections, see: "Dynamically adding sections (cloning)" on page 440.

Cloning a section based on the number of records in a detail table

This script creates as many clones of a section as there are records in a detail table. It assigns the new sections a unique name.

```
var printSections = merge.template.contexts.PRINT.sections;
var numClones = record.tables['detail'].length;
for( var i = 0; i < numClones; i++){
       var clone = printSections["Section 1"].clone();
       clone.name = "my_section_clone_" + i;
       printSections["Section 1"].addAfter(clone);
}
```

Cloning a section based on data and assign a background PDF

This script clones a section based on data fields. It disables the source section first and then calls the `addPolicy()` function. `addPolicy()` clones the section, renames it and sets a PDF from

the resources as its background. It explicitly enables the clone and then adds it to the Print context.

```
var printSections = merge.template.contexts.PRINT.sections;
merge.template.contexts.PRINT.sections["Policy"].enabled = false;
if(record.fields.policy_a == 1) {
	addPolicy('a');
}
if(record.fields.policy_b == 1) {
	addPolicy('b');
}
function addPolicy(policy){
	var resourceUrl = 'images/policy-' + policy + '.pdf';
	var clone = printSections["Policy"].clone();
	clone.name = "policy_" + policy;
	clone.background.url = resourceUrl;
	clone.enabled = true;
	printSections["Policy"].addAfter(clone);
}
```

margins

The `margins` object is used to set a Print section or Master Page's margins in a Control Script (see "Control Scripts" on page 429 and "Control Script API" on page 931).
Note that Master Pages are only used in a Print context (see "Master Pages" on page 156).

The `margins` object is retrieved via the `section` object (see "section" on page 978) or via the `masterpages` array in a template (see "masterpage" on page 934), respectively.

Fields

Field	Type	Description
bottom **left** **right** **top**	String	These fields allow to set the bottom, left, right and top of a Print section, using a Measurement string, for example: "2in" (this sets a margin to two inches).
footer **header**	String	These fields allow to set the header and footer of a Master Page, using a Measurement string.

Examples

Setting the margins of a Print section

```
var sectionA = merge.template.contexts.PRINT.sections["Section A"];
sectionA.margins.top = "2in";
sectionA.margins.left = "2in";
sectionA.margins.right = "2in";
sectionA.margins.bottom = "2in";
```

Setting the header and footer of a Master Page

```
var masterA = merge.template.masterpages["Master page A"];
masterA.margins.header = "2in";
masterA.margins.footer = "2in";
```

paginate()

This method of the `section` object (see "section" on page 978) triggers pagination of the current section. The pagination process re-establishes page boundaries, updates page numbers and page counts, and reapplies Master Pages.
Depending on whether page numbering restarts in each section this may affect the page numbers in other sections as well (see "Configuring page numbers" on page 152).
When the pagination process has ended, the script resumes.

The 'current section' is always a Print section, since this method can only be used in Post Pagination Scripts, and Post Pagination Scripts only run on the Print context (see "Post Pagination Scripts" on page 445).
You only need to call `merge.section.paginate()` in a Post Pagination Script if the script has added or removed content to such an extent that the page boundaries need to be renewed.

For an example see: "Creating a Table Of Contents" on page 447.

Pagination

The Pagination object that is accessed via the `section` object (see "section" on page 978) contains the total page count, sheet count and start/end page numbers for a **single section**.
These properties are read-only and are only available in Post Pagination Scripts (see "Post Pagination Scripts" on page 445).

For the total page and sheet count of **all Print sections** together use the `merge` object's "Pagination" on page 991 (see also: "merge" below).

For information about where an element is located and the counts of pages and sheets in a **page bundle** - a group of sections in which the page numbering continues - use "info()" on page 977.

Field	Type	Description
pageCount	Number	The total number of pages in this section.
pageEnd	Number	The index (1-based) of the last page in this section.
pageStart	Number	The index (1-based) of the first page in this section.
sheetCount	Number	The total number of sheets in this section.
sheetEnd	Number	The index (1-based) of the last sheet in this section.
sheetStart	Number	The index (1-based) of the first sheet in this section.

merge

In Control Scripts, the root level instance of the object `merge` is the entry point from where you can query and change the way contexts are merged. It gives access to the template with all its contexts and sections.
For more information about Control Scripts, see "Control Scripts" on page 429 and "Control Script API" on page 931.
Some of the objects are also useful in Post Pagination Scripts; see "Post Pagination Scripts" on page 445 and "Post Pagination Script API" on page 967.
For sample scripts, follow the links to the respective objects.

Field	Type	Description
channel	"Channel" on page 964	The final output channel: EMAIL, PRINT or WEB. The channel doesn't change when the output consists of different contexts. When generating an email, for example, the channel is EMAIL, even when merging

Field	Type	Description
		the Print context to attach it to the email.
"context" on page 969	Context	The context rendered by this merge run. If for one record, different contexts need to be output (for example, when the Print context is attached to an email) a record is merged multiple times: once per context. Per merge run, `merge.context` shows with which context the record is merged.
pagination	"Pagination" on the next page	Contains the total page count and sheet count of all sections in the Print context after pagination.
"section" on page 978	Section	In Standard Scripts, this object defines the section that is being merged. **Note!** In Control Scripts, `merge.section` is not available. To retrieve any section in a Control Script, use: `merge.template.contexts.ContextType.Section['Section name'];` (for example: `merge.template.contexts.PRINT.sections["Section EN"]`). In Post Pagination Scripts, only Print sections are available.
"template" on page 960	Template	This object contains the template and all of its contexts. It can be used to find out which contexts are available in the template, using `merge.template.contexts` (see "context" on page 969) and to manipulate the sections in those contexts (see "section" on page 978).

Pagination

The Pagination object that is accessed via the `merge` object (see "merge" on page 989) contains the total page count and sheet count of the **entire Print context** after pagination. These properties are read-only and are only available in Post Pagination Scripts (see "Post Pagination Scripts" on page 445).

For information about where an element is located and the counts of pages and sheets in a **page bundle** - a group of sections in which the page numbering continues - use "info()" on page 977.

For the total page count, sheet count and start/end page numbers of a **single section**, use the `section` object's "Pagination" on page 988 (see also: "section" on page 978).

Field	Type	Description
pageCount	Number	The total number of pages in all Print sections together.
sheetCount	Number	The total number of sheets in all Print sections together.

Generating output

When merged with a record set, the templates made in the Designer can generate two types of output: Print and Email.

Print output

Connect supports a number of different types of print outputs. These include:

- AFP
- IPDS (spool file created by Connect, with printer control available through the "Print Manager" on page 1029)
- PCL
- PDF
- PostScript (including the PPML, VIPP and VPS variants)

Print templates (also called Print *sections*), are part of the Print context. They are meant to be printed directly to a printer or a printer stream/spool file, or to a PDF file (see "Generating Print output" on the next page).
The Print context can also be added to Email output as a PDF attachment; see "Generating Email output" on page 1017.
When generating output from the Print context, each of the Print sections is added to the output document, one after the other in sequence, for each record.

To dynamically select a section for output, use a Control Script; see "Control Scripts" on page 429.

There is a number of settings in the Print context and Print sections that have an impact on how the Print context is printed; see "Print settings in the Print context and sections" on page 134.

To split the Print output into several files, see "Splitting printing into more than one file" on page 1007.

Email output

The Email context outputs HTML email with embedded formatting to an email client through the use of an email server. The HTML generated by this context is meant to be compatible with as many clients and as many devices as possible.

Although the Email context can contain multiple Email templates, only one of them can be merged with each record. Which one is used, depends on a setting; see "Email output settings in the Email context and sections" on page 1019.

Email is sent in a single batch for the whole record set.

To test a template, you can test the scripts (see "Testing scripts" on page 403) and send a test email first (see "Send (Test) Email" on page 543), before actually sending the email (see "Generating Email output" on page 1017).

Attachments

Output, generated from an Email template, can have the following attachments:

- The contents of the Print context, in the form of a single PDF attachment. (Compression options for PDF attachments can be specified in the Email context's properties; see "Compressing PDF attachments" on page 176.)
- Other files, an image or a PDF leaflet for example.

Attaching the Print context is one of the options in the "Send (Test) Email" on page 543 dialog.

To learn how to attach other files, see "Email attachments" on page 187.

Generating Print output

Connect supports a number of different types of print outputs. These include:

- AFP
- IPDS (spool file created by Connect, with printer control available through the "Print Manager" on page 1029)
- PCL

- PDF
- PostScript (including the PPML, VIPP and VPS variants)

> **Note**
>
> The maximum number of pages is 9999 per record for each Print section.

Print output can only be generated from the Designer when a data set is available (see "Loading data" on page 341). The Designer merges all sections in the Print context (see "Print context" on page 132) with the data set, and generates the output using those data values.

To generate Print output, select **File** from the menu and choose **Print**.

- **File > Print...** allows the following printing options:
  - Using the **Default** output settings.
    For more details, see "Print using standard print output settings" on page 996
  - Using the same settings that were **last used** to produce printed output.
    For more details, see "Print using standard print output settings" on page 996
  - Using **entirely new output settings** set via the Advanced option, which allows selection from a myriad of print output options.

    > **Note**
    >
    > These settings cannot be saved for later re-use. To do that, one should instead create printing Presets, which are designed to allow just this behaviour.

    For a detailed description see "Print using Advanced Printer Wizard " on page 1005.
  - Using previously saved **Printing Preset** options.
    See "Print Presets" on page 998 for more details.

Saving Printing options in Print Presets

Selecting **File > Print Presets** allows you to create or modify printing Presets (which contain all the printing options), which can be saved for re-use in later print runs. This can be particularly handy when creating special print runs, that need to be run periodically.

These presets make it possible to do such things as filtering and sorting records, grouping documents and splitting the print jobs into smaller print jobs, as well as the more standard selection of printing options, such as binding, OMR markings and the like.

See "Print Presets" on page 998 for more details.

Print settings in a template

There are a number of settings for the Print context and Print sections that have an impact on how Print sections are printed, which **cannot** be made in the Print Wizard or influenced through either a Job Creation Preset or an Output Creation Preset. They are made in and saved with the template.

These settings are:

- **Duplex printing**. Duplex printing has to be enabled for a Print section, in order to print that section on both sides of the paper. The same applies to Mixplex printing.
  See "Enabling double-sided printing (Duplex, Mixplex)" on page 145.
- **Finishing**. The Print context , as well as each of the Print sections, can have its own **Finishing** settings. In printing, Finishing is the way pages are bound together after they are printed.
  See "Setting the binding style for the Print context" on page 135 and "Setting the binding style for a Print section" on page 145.
  Also see "Finishing options" on page 691 for an explanation of the Finishing options.
- **Bleed**. The margins *around* a page are called the Bleed. It can be used on some printers to ensure that no unprinted edges occur in the final trimmed document.
  See "Page settings: size, margins and bleed" on page 148.
- **Black overprint**. The option to print small black text over other colors is referred to as black overprint. See "Overprint and black overprint" on page 136.

Aborting content creation

You may want the content creation process to be aborted in certain situations; for example, when a template script fails to load remote content. To abort the content creation process, you may raise a fatal error from within a script in the template; see "fatalError(message)" on page 815.

When a script calls this function in Preview mode, the script that triggers it is marked with an error icon in the Scripts pane, and the given message is displayed in a hint.

When generating output from the Designer, the Designer will log the error and display an error dialog with the given message. Content creation is aborted.

Print using standard print output settings

When using either the **File > Print...** or the **File > Proof Print...** option, the Print Configuration dialog appears. This dialog allows you to print the template using **Default** printer settings, or the **Last Used** printer settings or by using previously created Printing Presets.

To learn how to create Printing Presets please see Job Creation Presets and Output Creation Presets.

- **Configuration Selection** Group:
  - **Output Creation**: Leave the selection at the Default or use the drop-down to select an existing Output Creation Preset.
    Use the Gear button to edit the currently selected Preset or to reload the list of Presets from the system.
  - **Job Creation**: Leave the selection at the Default or use the drop-down to select an existing Job Creation Preset.
    Use the Gear button to edit the currently selected Preset or to reload the list of Presets from the system.
  - **Preset Summary**: Displays a summary of the settings for the currently selected Presets (Default or created).

> **Note**
>
> The Default output type (PDF Output) is actually a built in system Preset, whilst the Last Used settings can likewise be considered an un-named and un-saved Preset.

- **Records** Group:
  - **All**: Outputs all records in the active dataset.
  - **Selection**: Allows selection of a range of records or a custom selection.
    You can specify individual records separated by semi-colons (;) or ranges using dashes.
    For example: 2;4;6-10 would print pages 2, 4, 6, 7, 8, 9 and 10.
  - **Apply filtering and sorting to record selection** checkbox: Check o filter and/or sort records. Selecting this will open both the "Data filtering options" on page 689 and "Sorting options" on page 693 pages.
- **Copies** Group :
  - **Copies**: Enter the number of output copies you want.
  - **Collate**: When printing multiple copies you can check this checkbox to have the record copies printed together.
    For example in a three record job the records would print out as 1-1-2-2-3-3, rather than 1-2-3-1-2-3.

Wizard navigation buttons

- **Advanced** button: Click to open the "Print using Advanced Printer Wizard " on page 1005where you can manually change the printing options.

> **Note**
>
> Any settings made within the **Advanced Print Wizard** do not permanently update any Preset(s) being used.

- **Print** button: Click to produce print output according to the current settings.
- **Cancel** button: Cancels the Print Wizard, without creating any printout.

Print Presets

Print Presets are files that contain **settings for a print job and printer**. They allow you to re-use your job and printer settings.

All of PrintShop Mail Connect's print job and printer options are also available in the Advanced Printer Wizard, which appears when you select **File > Print** or **Proof Print** in the menu and then click the **Advanced** button, but the wizard cannot save your settings.

Select **File > Print Presets** to create or modify a Print Preset.

Print Preset types

There are two types of Print Presets: "Job Creation Presets" on the next page and "Output Creation Presets" on page 1001. The settings in each are used in a different phase of the print process.

- A **Job Creation Preset** is used by the Connect Server after a template and data have been merged. Based on the settings in a Job Creation Preset the server can filter, sort, and group the print content items, add meta data and make finishing settings.
  The resulting print job is then handed to the Weaver engine for output creation (see also: "Connect: a peek under the hood" on page 102). This engine generates the final print output file, or files.
- The settings in an **Output Creation Preset** define the destination and all the appropriate print options and printer settings.

How to use Print Presets

Print Presets are not bound to a particular template. They can be used with different Print templates, as long as the data contains the required information.

Job Creation Presets and Output Creation Presets can also be used in different combinations, although not randomly:

- Groups of documents, such as invoices for a specific customer, can be printed separately. This requires that the documents are grouped in advance. In other words, the Separation settings in the Output Creation Preset and the Grouping settings in the Job Creation Preset must match. (See also:"Splitting printing into more than one file" on page 1007.)

- Conditions and output file names in an Output Creation Preset can only make use of those meta data tags that have actually been added to the print job via the Include meta data setting in the Job Creation Preset.
- The finishing options selected in the Job Creation Preset should match the capabilities of the printer selected in the Output Creation Preset.

To apply Print Presets in the **Designer**, select **File > Print** or **Proof Print** and then select an Output Creation Preset and (optional) Job Creation Preset from the drop-down lists in the Print dialog. It will be indicated if the current data doesn't contain the fields required by the selected Job Creation Preset.

Job Creation Presets

A **Job Creation Preset** is a file that contains settings for the creation of a print job. It is used after a template has been merged with data, but before the actual print output file is produced. Merging a template with data results in a set of **print content items**. Based on the settings in a Job Creation Preset, the Connect server can take a number of actions on a set of print content items. The actions are described below.

In addition, **runtime parameters** can be defined in a Job Creation Preset. Filled with a value at runtime, these can be used for comparisons against conditions within the Job Creation, or in external sorting programs.

Select **File > Print Presets > Job Creation Settings...** to run the "Job Creation Presets Wizard" on page 684.

> **Note**
>
> The Job Creation options are also available in the Advanced Printer Wizard, which appears when you select **File > Print** (or **Proof Print**) and click the **Advanced** button. However, the Advanced Printer Wizard cannot save your settings to file for re-use.

Filter and sort

While creating a print job, the Connect server can filter out certain print content items based on the conditions defined in a Job Creation Preset (see "Data filtering options" on page 689). The conditions are evaluated per record and can be based upon such diverse criteria as the value of data fields, the value of runtime parameters (see "Runtime Parameter Options" on page 686), the binding options used in a source document, or its size.

Note that the *overriding* finishing options, which can also be defined in a Job Creation Preset, are not taken into account while filtering. First, the source documents are filtered. Any overriding finishing options are applied at a later stage.

The remaining items can be sorted, either by the Connect Server, based on data fields, or by some external sorting software (see "Sorting options" on page 693).

> **Tip**
>
> External sorting benefits considerably from runtime parameters, defined in the Job Creation Preset (see "Runtime Parameter Options" on page 686).

Grouping

While creating a print job, the Connect Server can group Print content items into Jobs, Job Segments and Document Sets (see "Grouping options" on page 696).
This allows to organize content in a certain way, for instance, to gather all foreign mail pieces into one job segment, or to commingle multiple sets of Print content items in one job.

The existence of groups in a print job also allows the Weaver engine, in the next phase of the print process, to separate the output accordingly into different (spool) files (see "Separation" on page 1003).
It is always possible to separate the output on the document level, i.e. output individual documents as separate print files.
In order to split the output on other levels, however, the Print content items must be grouped first.

Include meta data

Meta data can be added to a print job at the Job, Job Segment, Document, Document Set and Page level (see "Meta Data options" on page 699).

In the next phase of the print process these meta data tags can be used in **output file names** and in various **conditions**.
If the output format is PDF/VT, the meta data will also be included in the print output itself. For AFP, this is optional.

If runtime parameters are defined in the Job Creation Preset (see "Runtime Parameter Options" on page 686), their value can be used as meta data.

Override finishing options

The Print context and Print sections can have their own finishing options (see "Print settings in the Print context and sections" on page 134.) A template could be setup with a glue binding, for example.
When the printer doesn't support the chosen finishing options you could override the finishing options, and use staple binding, for instance.
More importantly, the Override finishing options option allows you to make **dynamic** finishing settings, and this not only on the Section level but also on the Document, Document Set and Job Segment level.

If runtime parameters are defined in the Job Creation Preset, the passed information can be used in conditions (see "Runtime Parameter Options" on page 686).

Output Creation Presets

An **Output Creation Preset** is a file that defines the printer model and output type, and all desired **print options** and **printer settings**. It is used by the Weaver engine (see "Connect: a peek under the hood" on page 102), which generates the final print output file (or files).

The input of the Weaver engine is a print job, assembled by the Connect Server. The set of print content items in the print job may have been filtered, sorted and grouped. The server may have added meta data and finishing settings to the job, according to the instructions in a Job Creation Preset (see "Job Creation Presets" on page 999).

Select **File > Print Presets > Output Creation Settings...** to run the "Output Creation Presets Wizard" on page 700.

> **Note**
>
> The Output Creation options are also available in the Advanced Printer Wizard, which appears when you select **File > Print** (or **Proof Print**) and click the **Advanced** button. However, the Advanced Printer Wizard cannot save your settings to file for re-use.

Printer model and output type

An Output Creation Preset specifies the type of printer the output should be suitable for, and determines where that output should then be sent: to a folder, LPR queue or Windows printer (see "Print options" on page 704).

Printer definitions

A printer model and its capabilities for Connect are defined in a printer model definition file (.OL-printerdef). Connect comes with a number of such files. However, there are so many different printer models with their own settings and capabilities that it is simply impossible to provide definition files for all of them.
If there is no definition file for your PostScript printer you may create a customized printer definition file yourself using the PostScript Printer Definition (PPD) file provided by the printer manufacturer (see "Dynamic PPD Options" on page 788).
Other customized Printer Definitions can be obtained from Upland OL Support. They need to be tailor-made based on the information that you provide about both your printer and your specific needs.

Certain printer definition files will work with several types of printers (consecutive versions of a certain printer model, for example). Note that those printers may still have different capabilities. When creating or selecting Print Presets, always consider the actual capabilities of your specific printer.

Mapping media types to printer trays

An Output Creation Preset can map media types to printer trays (see the "Printer settings" on page 786 dialog). This option is available for AFP, IPDS, PCL, PPML and PostScript printers that are configured for cut-sheet printing.

Print virtual stationery

The virtual stationery in a Print template (also called "Media" on page 159) is in fact an image that is normally only visible in the Designer in the background of the template to make designing a template easier.
If indicated in the Output Creation Preset, the image will also be printed. The output can then be printed on plain paper instead of pre-printed paper. Printing virtual stationery is also very useful when producing PDF output for digital viewing.

Imposition

An Output Creation Preset can tell a printer to print multiple pages on a single sheet (Simplex, or Duplex). This is called **Imposition**, also known as **N-Up** printing.

The two Imposition types are Cut & Stack and Stack by Column (see "Imposition options" on page 774).
With each Imposition type there is a wide range of options to control the exact layout of the output: the number of pages that go on one sheet, their order, the way they are stacked, the stack depth, the margins, markings, etc.

Note that when Imposition is used, the possibilities to separate the output (see "Separation" below) and to add additional content ("Additional content" on the facing page) are limited. Groups and meta data on lower levels than the Job Segment level (as defined in a Job Creation Preset) are no longer available after the Imposition process.

Booklet Imposition

Booklet Imposition defines how to generate booklets in the output (see "Booklet Options" on page 773).

As printing booklets implies printing multiple pages on a single sheet, this is used in conjunction with some Imposition settings.

Separation

The print output can be split - 'separated' - into discreet portions. An Output Creation Preset can either separate documents, document sets, job segments or jobs, or it can split the output between *counts* of sheets, documents, document sets or job segments. In addition the preset can tell the printer to add slip sheets and/or jog the output. See "Separation options" on page 807.

Using **document sets**, **job segments** or **jobs** in the Separation settings requires that documents in the print job be grouped that way in advance, following the instructions in a Job Creation Preset (see "Grouping" on page 1000).

Note

Separating the output using Grouping on levels lower than the Job Segment is not possible when you use Imposition.
With Booklet Imposition, however, the output can be separated on all levels.

Additional content

Via an Output Creation Preset, content (**Text**, **Images**, **Barcodes** and **OMR Marks**) can be added to the output at the time of output creation. The additional content can be either static or variable (see "Additional Content" on page 732).

Additional content is particularly useful when you might need to drive custom processes on production machines using either Barcodes or OMR Marks, or if you need to add some last minute additions to the print job via text and/or images.
It helps to keep print production specific additions out of the templates, so templates don't have to be changed if production equipment or postal delivery requirements change.

In additional content you may use the meta data that was included during job creation (see "Include meta data" on page 1000).

Note

When you combine **Imposition** with additional content, the content needs to be added to every page - not once per sheet - in order to have access to the meta data on all levels. If the option **Output once per sheet** is checked, only the meta data at the Job Segment level will be available.

Inserter marks

Inserter marks can be added to the print output in accordance with the settings in an Output Creation Preset (see "Inserter options" on page 783). The available options are dependent on the selected High Capacity Feeder (HCF) model. These machines are also commonly referred to as Inserters or Folder-Inserters.
If no HCF file is available for a particular inserter machine, adding OMR Marks and barcodes as Additional Content is an alternate way to drive an inserter (see "Additional content" above).

Print using Advanced Printer Wizard

The **Advanced Printer Wizard** allows you to select from any and all output settings.

The Wizard can be used to generate once-off print runs (either entirely from scratch, or based upon selected pre-existing Presets).

Note: These print runs cannot be saved as presets and can only be replicated in the following print run, using the **Last Used** option.

The output settings are determined by selections made throughout the Wizard. For example, if you want to add Inserter Marks to the output, you select the Add Inserter Marks option on the first page of the Wizard, and the Inserter Options page will then appear later in the Wizard.

The first page of the Advanced Printer Wizard is the "Print options" on page 704 page.

Adding print output Models to the Print Wizard

Connect comes with several pre-prepared print output Models. These include Advanced Function Presentation (AFP), Intelligent Printer Data Stream (IPDS), Printer Control Language (PCL), Portable Document Format (PDF) and PostScript (including the PostScript variants of PPML, VIPP and VPS).

To keep the Print Wizard interface manageable only a limited range of print output Models are available by default. Additional print output Models can be added to the list at any time, though. They can be selected from the range of pre-prepared Models that come bundled with Connect, or from tailored Models prepared for you by Upland OL, or from PostScript printer Models that you define yourself, through importation of the PostScript Printer Definition (PPD) file.

The following topic describes how to add new Models to the Print Wizard. These Models will always be available in the Print Wizard thereafter, unless deliberately removed.

How to add print output Models within the Print Wizard

Here is how you can add print output Models to the Print Wizard from within the Print Wizard dialog itself.

1. Select **File > Print...** from the menu. The Print dialog will be launched.
2. Click on the **Advanced** button. The Print Wizard will be launched.

3. Click the settings button at the end of the Model selection list box.
   Select from one of the options:
   1. Select **Import Definition** to add a customized Printer Definition. These are customized files created by Upland Objectif Lune in conjunction with you. They will have been customized specifically for your printer(s).
   2. Select **Create Definition from PPD** to add a customizable PostScript Printer Model using the printer manufacturer provided PostScript Printer Definition (PPD) file.
      This launches the **PPD Import Wizard**. In the Wizard, do the following.
      - Select the printer's PPD file. Once selected details about that specific printer will load in the information box, for confirmation.
        NOTE: The PPD information extraction and processing can take some seconds.
      - If you are satisfied that the printer information is correct, then press the **Finish** button to complete the importation process.
        This will then launch another dialog to allow setting the new Pinter Model name.

      Once a PPD has been selected, then the "Dynamic PPD Options" on page 788 page will be presented at some point in the Print Wizard. When it appears is dependant upon other choices made in the Wizard, This page provides the functionality for assigning printer specific options to the Connect Template.

      See "Dynamic PPD Options" on page 788 for more details.
   3. Select **Edit available printers** to add standard Printer Definitions. These are the standard print outputs installed with Connect.
      - In the **Preferences** dialog that launches, select the print output model(s) to be added to the Print Wizard, then click OK.
   4. Select **Export printer definition from output preset** to extract a Printer Definition from a new or modified Printer Model. This launches the Save Printer Definition Dialog.

How to add print output models from within the Designer

Here is how to add print output Models from within the main Designer interface itself.

1. Select **Window > Preferences...** from the menu. Preference dialog is launched.
2. Select **Print > Available Printers** from the options.
3. In the **Available Printers** area, select the print output options to be added to the Print Wizard, then click OK.

Splitting printing into more than one file

By default, when Connect saves the print output spool file to a directory, it creates one spool file that contains all the generated documents (one document per data record). It is, however, possible to output one spool file per document, or to create groups of documents and store those in separate spool files. It is also possible to split documents that are longer than a certain number of pages into multiple output files. This topic explains how to do that.

Splitting one document into multiple files

If a *document* - the output of one *record* - can have too many pages so that it cannot fit into one envelope, it can be split automatically into equal parts using the **Repeat sheet configuration** option in the Sheet Configuration dialog. For instructions see "Applying a Master Page to a page in a Print section" on page 158 and "Sheet Configuration dialog" on page 546.

Grouping documents

Documents in a print job can be grouped on three levels: Job, Job Segment, and Document Set, via a **Job Creation Preset** (see "Job Creation Presets" on page 999).
For instance, in a mailing destined for recipients in both Canada and the United States, you might want to group the documents by country (Job level) in order to separate the US and CA recipients. You could further sort the mail pieces by state/province (Job Segment level) and then by individual postal codes (Document Set level).

Creating separate output files

To make **each document** or **groups of documents** go into a separate file, a print job needs to be 'separated'. *Separation* is one of the options to set in an **Output Creation Preset** (see "Output Creation Presets" on page 1001). An Output Creation Preset also determines where the output will go.

For example, if a mailing has two groups on the Job level: one of recipients in Canada and one of recipients in the United States, separating the output on the Job level and printing to PDF would result in two PDF files.
If the same documents were also grouped by state/province on the Job Segment level, then splitting the job at the Job Segment level would result in one file per state/province.

Naming output files

When output is split into multiple files, each file probably needs to get a name that identifies the document or group of documents inside it.
For example, if documents are grouped by state or province on the Job Segment level, and the job is split at the Job Segment level, the output files probably need to be named after a state or province.
Here's how to do that:

1. First, define **meta data** at the intended separation level. This is done in the Job Creation Preset (see "Include meta data" on page 1000).
2. Then, in the Output Creation Preset, use those meta data in the Job Output Mask (see "Output options" on page 706). They are inserted into the file name as variables. For example: `${segment.metadata.State}` refers to a meta data tag `State` defined on the job segment level.
   For the complete list of output variables, see "Print output variables" below.

   > **Note**
   >
   > Only meta data defined on the actual separation level are accessible to the Job Output Mask.

Print output variables

In Print output, File name variables can be used to create dynamic output file names, while Content variables can be used in additional content and in the Conditional field when selectively adding inserts in the "Inserter options" on page 783.

Note

When using a variable in a condition, do not wrap it in `${ }`. This notation is only necessary in the context of a text (like file names or additional page content).

File name variables

File name variables are available in a few places in the "Print options" on page 704:

- In the **Job Output Mask** and **Job Output Folder** fields when using the Directory option.
- In the **Job Name** field when using the Windows Printer option.

Tip

You can name the output file or folder through the Workflow configuration.

This is done by setting a Job Info variable in the Workflow configuration see Job Info variables in the Workflow Online Help) and then referencing it in the "Print options" on page 704 fields via the variable `@automation.JobInfoX@` (where "X" is the number of the Job Info variable).

If the output is to be separated into multiple files, some Output content variables can also be used as Output file variables, depending on the level of separation (see "Content variables" on page 1012).

Template

${template}	`${template}` is a shorthand for `${template.base}_${template.nr,0000}.${template.ext}`. It expands to a name based on the template name. A four digit sequence number is added at the end of the base name. The file extension is determined by the selected output format. The `0000` in `${template.nr,0000}` is a format pattern that takes care of formatting the number with at least four digits and leading zero's. See "Formatting date and number values " on page 1016, below.

	Example If the template file is `C:\Data\My-Invoices-EN.OL-template` which gets printed to PDF, then `${template}` expands to `My-Invoices-EN_0001.pdf`.
${template.base}	Returns the base name of the template, which is the name of the template file without its path and without the trailing file extension. **Example** If the template file is `C:\Data\My-Invoices-EN.OL-template`, then `${template.base}` expands to `My-Invoices-EN` .
${template.name}	Returns the name of the template file without the path. **Example** If the template file is `C:\Data\My-Invoices-EN.OL-template`, then `${template.name}` expands to `My-Invoices-EN.OL-template`. Note that `${template.name}` still includes the extension of the template file (.OL-template in the example above).
${template.nr}	An automatic sequence number belonging to the current output file. It is automatically incremented for each new output file that gets created when Separation has been selected in the Output Creation Preset. It is possible to format the number using a pattern and locale. See "Formatting date and number values " on page 1016, below.
${template.ext}	The extension that corresponds to the chosen output format (in lower case). For example, for PDF output, `${template.ext}` would be **pdf**, for PostScript output, `${template.ext}` would return **ps**. Note that `${template.ext}` does not include a leading dot.

File

${file}	`${file}` is a shorthand for `${file.base}_${file.nr,0000}.${file.ext}`. It expands to an internally generated file name with a four digit sequence number at the end. The file extension is determined by the selected output format. The `0000` in `${file.nr,0000}` is a format pattern that takes care of formatting the number with at least four digits including leading zero's. See "Formatting date and number values " on page 1016, below. **Example** When printing to PDF, `${file}` could expand to **merged.5852941188491153960_0001.pdf**.
${file.base}	Expands to a unique, internally generated file name, without a trailing dot or extension.
${file.ext}	The extension that corresponds to the chosen output format, for example **pdf** or **ps**. Note that `${file.ext}` does not include a leading dot.
${file.nr}	An automatic sequence number belonging to the current output file. It is automatically incremented for each output file that gets created within the same job. It is possible to format the number using a pattern and locale. See "Formatting date and number values " on page 1016, below.
${file.pageCount}	This variable was introduced for use in Printer Definitions for PostScript printers. As it entails page buffering and could easily lead to Out of Memory errors with big jobs, **usage of this variable in an Output Preset or in the Print Wizard is discouraged; it should be regarded as deprecated.**

Job

${job}	`${job}` is a shorthand for `${job.base}_${job.nr,0000}.${job.ext}`. It expands to a name based on the name of the applied Job Creation Preset (or 'Untitled' if no Job Creation Preset was used). A four digit sequence number is added at the end of the base name. The file extension is determined by the selected output format. The `0000` in `${job.nr,0000}` is a format pattern that takes care of formatting the number with at least four digits including leading zero's. See "Formatting date and number values " on page 1016, below.
${job.base}** or **${job.name}	Returns the name of the applied Job Creation Preset without any extension. Expands to **Untitled** if no Job Creation Preset was used.
${job.nr}	An automatic sequence number belonging to the current output file. It is automatically incremented for each new output file that gets created. Note, that multiple output files are created, for example, when output separation has been selected for output creation. It is possible to format the number using a pattern and locale. See "Formatting date and number values " on page 1016, below.
${job.ext}	The extension that corresponds to the chosen output format, for example **pdf** or **ps**. Note that `${job.ext}` does not include a leading dot.

Content variables

The variables listed below can be used in text, barcodes, and OMR and Image data in the "Additional Content" on page 732 page, and in the Conditional field when selectively adding inserts in the "Inserter options" on page 783.

If the output is grouped and separated, Content variables on the separation level and above are **also available as File name variables**. For example, if the output is grouped on the job segment and document set level, and is to be separated on the Document Set level, the **set** and **segment** variables can also be used in the Job Output Mask field.

Warning

Use `count` variables with caution. They entail higher memory usage in Weaver (the engine that creates Print output). When, for example, `segment.count.pages` is used in additional text, the Weaver engine has to buffer all pages until it knows how many pages the segment counts in order to include that total in the additional text on each page. With big jobs this could easily lead to Out of Memory errors.

System

${system.time}	Displays the current system data and/or time. Can be formatted using the "Formatting date and number values " on page 1016, as seen below.

Page

${page.back}	True when the page is on the back of the sheet.
${page.front}	Boolean indicating whether the page is on the front of the sheet.
${page.height}	The page's height (in points).
${page.landscape}	True when the page's orientation is landscape.
${page.portrait}	True when the page's orientation is portrait.
${page.nr}** or **${page.sequence.document)	Page number in the document.
${page.sequence.sheet}	Page number on the sheet.
${page.sequence.set}	Page number within the document set.
${page.sequence.segment}	Page number within the job segment.

${page.sequence.job}	Page number within the job.
${page.width}	The page's width (in points).

Sheet

${sheet.count.pages}	Total number of pages on the sheet.
${sheet.duplex}	True when printing on the sheet is duplex.
${sheet.height}	The sheet's height (in points).
${sheet.pageDevice}	Array specifying the sheet's Type (e.g. Plain), Weight (e.g. 83), Color (e.g. yellow), Width (in points), Height (in points) and Tumble (e.g. false).
${sheet.nr}** or **${sheet.sequence.document}	Sheet number within the document.
${sheet.sectionName}	Name of the Template Section that the Sheet belongs to.
${sheet.sequence.set}	Sheet number within the document set.
${sheet.sequence.segment}	Sheet number within the job segment.
${sheet.sequence.job}	Sheet number within the job.
${sheet.simplex}	True when printing on the sheet is simplex (one-sided).
${sheet.width}	The sheet's width (in points).

Document

${document.metadata. ***propertyname*****}** or	Value of a meta data property of the document.

${document.metadata ['*propertyname*']}	The *propertyname* must have been defined as a Tag Name on the Document Tags tab of the "Meta Data options" on page 699 page in the *Advanced Print Wizard* or *Job Creation Preset*. Note: this variable is only available if Separation based on Document has been selected on the "Separation options" on page 807 page in the *Advanced Print Wizard* or *Output Creation Preset*.
${document.count.pages}	Total number of pages in the document.
${document.count.sheets}	Total number of sheets in the document.
${document.nr}** or **${document.sequence.set}	Document number within the document set.
${document.sequence.segment}	Document number within the job segment.
${document.sequence.job}	Document number within the job.

Set

${set.metadata.*propertyname*}** or **${set.metadata ['*propertyname*']}	Value of a meta data property of the Document Set. The *propertyname* must have been defined as a Tag Name on the Document Set Tags tab of the "Meta Data options" on page 699 page in the *Advanced Print Wizard* or *Job Creation Preset*. Note: this variable is only available if Separation based on Document Set has been selected on the "Separation options" on page 807 page in the *Advanced Print Wizard* or *Output Creation Preset*.
${set.count.pages}	Total number of pages in the document set.
${set.count.sheets}	Total number of sheets in the document set.

${set.count.documents}	Total number of documents in the document set.
${set.nr}** or **${set.sequence.segment}	Document set number within the job segment.
${set.sequence.job}	Document set number within the job.

Segment

${segment.metadata.*propertyname*}** or **${segment.metadata['*propertyname*']}	Value of a meta data property of the job segment. The *propertyname* must have been defined as a Tag Name on the Job Segement Tags tab of the "Meta Data options" on page 699 page in the *Advanced Print Wizard* or *Job Creation Preset*. Note: this is only available if Separation based on Job Segment or *Split At Exactly n Sheets* has been selected on the "Separation options" on page 807 page in the *Advanced Print Wizard* or *Output Creation Preset*.
${segment.count.pages}	Total number of pages in the job segment.
${segment.count.sheets}	Total number of sheets in the job segment.
${segment.count.documents}	Total number of documents in the job segment.
${segment.count.sets}	Total number of document sets in the job segment.
${segment.nr}** or **${segment.sequence.job}	Job segment number within the job.

Formatting date and number values

Date and number values can be formatted using an optional pattern and/or locale.

Form	Description	Example	Result
${*expression*}	Do not format.	`${system.time}`	July 4, 2009 12:30:55 PM
${*expression,pattern*}	Apply pattern with system locale	`${system.time, yyyyMMdd-HH:mm:ss}`	20090704-12:30:55
${ *expression,pattern,locale*}	Apply pattern with the specified country locale	`${system.time, "dd MMMM yyyy", fr}`	19 décembre 2017
${*expression,,locale*}	Apply a default format with the specified country locale	`${system.time,,fr}>`	19 décembre 2017 12:30:55

It is possible to enclose the values of the pattern and locale in single or double quotes. This is required for including whitespace in a pattern, or when the ${expression} would otherwise be ambiguous.

At run-time, the output engine determines the type of the value yielded by the expression. If this is a number, a number pattern is expected. For date/time-like types, a date pattern is expected. When no pattern is specified, some default format is applied. For other types, it is not possible to specify a pattern or locale.

For patterns and pattern characters see "Date and time patterns" on page 822 and "Number patterns" on page 844.

Generating Email output

The Email context outputs HTML email with embedded formatting to an email client through the use of an email server. The HTML generated by this context is meant to be compatible with as many clients and as many devices as possible.

Email is sent in a single batch for the whole record set.

To test a template, you can send a test email first.

Output, generated from an Email template, can have the following attachments:

- The contents of the Print context, in the form of a single PDF attachment. (Compression options for PDF attachments can be specified in the Email context's properties; see "Compressing PDF attachments" on page 176.)
- Other files, an image or a PDF leaflet for example.

Attaching the Print context is one of the options in the "Send (Test) Email" on page 543 dialog.

For more information see "Email attachments" on page 187.

Before generating Email output

- Decide on the use of an Email Service Provider; see "Using an ESP with PrintShop Mail Connect" on page 1021.
- Make sure that a data set is loaded, that any necessary files, such as images and attachments, are in place, and that the correct settings are selected (see below).
- If you're using style sheets to style the email, choose whether the styles must be added to the header of the email or to inline style properties as if local formatting was applied (this is also called "embedded CSS"). This setting is made per Email section. See "Email section properties" on page 540.
- You may want to **rasterize** certain elements, to ensure that most email clients would actually see the output. Rasterizing converts the element to a JPG or PNG image.
  To rasterize an element, right-click it and select **Rasterize options**. For a JPG image you can set the quality of the resulting image in a percentage.

> **Note**
>
> Rasterization options are only available for Boxes (elements); see "Boxes" on page 256.

> **Note**
>
> A business graphic in an Email section is rasterized by default and output as PNG image, because email clients usually don't support SVG images.
> SVG images in an Email section give an error in the Preflight window (see "Doing a Preflight" on page 405).

- For Email output, **PNG** is the preferred image format.
  EPS, PDF, SVG and TIFF images in an Email section are automatically converted to PNG to ensure that they can be seen in the email client.
- Test and validate the output; see "Testing Email output" on the facing page.

Email output settings in the Email context and sections

The following settings for the Email context and Email sections have an impact on how the actual emails are sent.

- An Email To Script must be available in the template and refer to a valid email address; see "Email header settings" on page 180. If any record does not have a valid email, this record is skipped automatically when generating email output.

> **Note**
>
> When you send a test email, the Email To Script will not be used; instead, the email will be sent to the address that you specify in the Send Test Email dialog.

- The sender(s), recipient(s) and the subject can be set using Script Wizards; see "Email header settings" on page 180.
- Default SMTP settings can be set in the preferences; see "Email header settings" on page 180.
- If there are multiple Email sections, only one of them can be merged with each record. Make sure that the correct section has been set as the default; see "Setting a default Email template for output" on page 180.
  To dynamically select a section for output, use a Control Script; see "Control Scripts" on page 429.
  When printing from the Designer, the currently open Email section will be outputted.

- A plain-text version of the HTML is added to each email if this option is checked in the Email section's properties (see "Email section properties" on page 540). With new templates this is always the case.
- If style sheets are used to style the email, the styles can be added either to the header of the email or to inline style properties as if local formatting was applied (this is also called "embedded CSS"). See "Email section properties" on page 540.
- PDF attachments can be compressed to make the files smaller; see "Compressing PDF attachments" on page 176.

Testing Email output

It can't be guaranteed that content will look the same in all email clients, particularly when it comes to older client versions, so it is recommended to always test and validate the output by sending a test email.
An example of a tool that can facilitate testing to ensure a consistent look and feel is **Litmus**: https://litmus.com/email-testing.
If you have a Litmus account, you can enter your Litmus test address when sending a test email (see "Generating Email output" on page 1017).
To narrow down the field of potential email clients by finding out which email clients are most commonly used by your target audience you can use Litmus' analytics tool (see https://litmus.com/email-analytics).

To generate Email output :

1. Open a template with an Email context.
2. Load a data file or database compatible with this template. See "Loading data" on page 341.
3. On the **File** menu, click **Send Email** or **Send Test Email**. In the dialog that appears you can, among other things, attach the Print context or the Web context to the email. See "Send (Test) Email" on page 543 for a description of all the options. Finally, click OK.

> **Note**
>
> When you send a test email, the Email To Script will not be used; instead, the email will be sent to the address that you specify in the Send Test Email dialog. If you have a Litmus account, you can enter your Litmus test address. To make the test address appear by default, you can set the default test address in the Email

Preferences: select **Window > Preferences**, click the arrow next to **Email**, click **General** and type the test address next to **Email Test address**.

See also: "Testing Email output" on the previous page.

Aborting content creation

You may want the content creation process to be aborted in certain situations; for example, when a template script fails to load remote content. To abort the content creation process, you may raise a fatal error from within a script in the template; see "fatalError(message)" on page 815.

When a script calls this function in Preview mode, the script that triggers it is marked with an error icon in the Scripts pane, and the given message is displayed in a hint.

When generating output from the Designer, the Designer will log the error and display an error dialog with the given message. Content creation is aborted.

Using an ESP with PrintShop Mail Connect

An email service provider (ESP) is a company that offers email marketing or bulk email services.

This topic explains why and how to use an ESP with PrintShop Mail Connect

Reasons to use an ESP

These are a number of reasons why you would need an ESP:

- ESPs ensure a high deliverability, as most ESPs are whitelisted or approved by ISPs (Internet Service Providers) as legitimate email delivery service. So they help you to avoid having mail detected as spam.
- ESPs provide comprehensive tracking options to measure open rates and they log which links were clicked and by who. Typically this information is available via an online dashboard.

- Most ESPs provide Bounce Management options. They will stop sending messages to addresses that return a hard bounce and retry for soft bounces before removing that address.
- EPSs can handle unsubscribes and prevent accidental sends in the future.

Choosing an ESP

The first thing to do to use an ESP with PrintShop Mail Connect is to choose an ESP and create an account.

Mandrillapp.com, a popular ESP, used to have a free account but now requires a paid MailChimp account. Luckily there are plenty of alternatives that provide free accounts (often capped to a max number of emails per month and sometimes having throttled output).

PrintShop Mail Connect has been tested with: Mandrillap.com, SendGrid (easy user management), MailGun (nearly instant statistics) and MailJet (shows best performance on the free account).

Adding an SMTP Preset for an ESP

After creating an account, add a SMTP settings preset in PrintShop Mail Connect for the chosen ESP, via the Preferences dialog of the Designer (see "Email SMTP settings" on page 182).

Make sure **Use authentication** is checked, and put in your SMTP Username in the box below.

> **Note**
>
> Presets for different ESPs are already available in the list of default presets.

Sending an email with an ESP

To send an email or test email with the use of an ESP, start generating the email as usual (see "Generating Email output" on page 1017). In the Send (Test) Email dialog, pay attention to the following settings:

- In the **Outgoing mail settings** area, select the preset for your ESP in the Presets drop-down.
- In the **Password** box, type the password provided by the ESP.

Note

The ESP might also have a test function you can use. Check the options of your ESP.

Adding custom ESP handling instructions

Most ESPs allow you to provide custom handling instructions as part of the email message, via custom headers. Typically these include instructions to enable open rate tracking, click through rate tracking and assign tags/categories to messages. Assigning a tag/category allows you to view statistics per email type in the dashboard of the ESP. Note that each ESP has its own notation and instructions.

In a Connect template, these custom headers can be added via the Email section properties (see "Meta information" on page 186).
Alternatively custom headers may be added through a Control Script (see "Control Scripts" on page 429, "Control Script API" on page 931 and "section" on page 978).
The following samples show how to assign a tag or category to a message for various ESPs.

SendGrid

Dashboard: https://app.sendgrid.com/

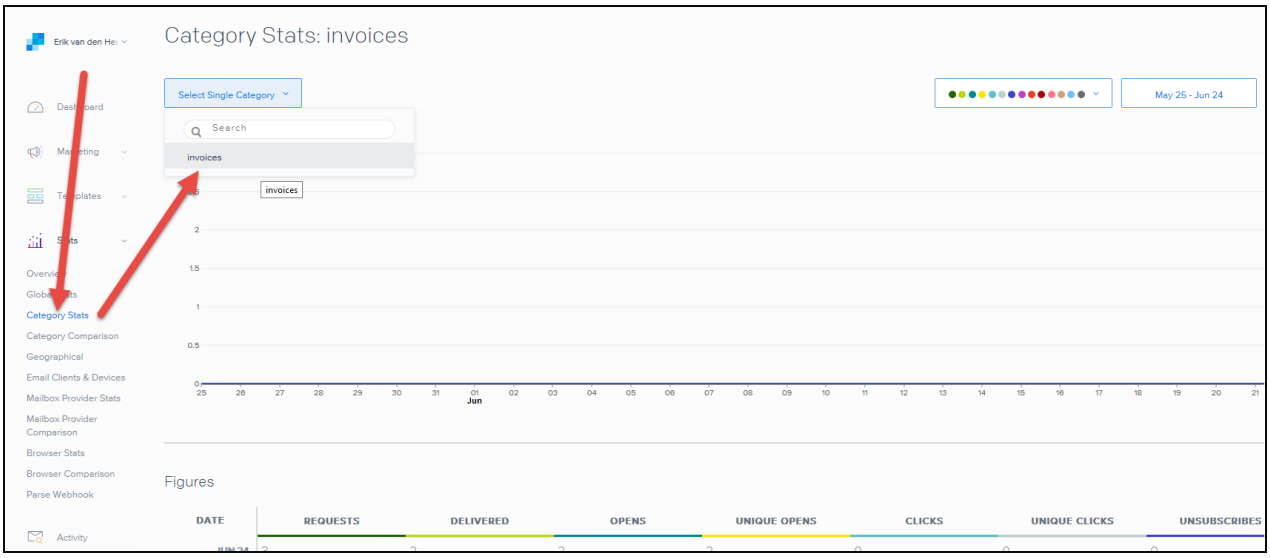
Documentation: https://sendgrid.com/docs/API_Reference/SMTP_API/using_the_smtp_api.html. (Use the SMTP API; the Connect Server sends SMTP email).

Add a *name* meta tag with the value *X-SMTPAPI* and set its content to the required JSON string (for example: *"'category': ['invoices']"*).
You can make this setting in the **Email section properties** (see "Meta information" on page 186).
Alternatively you could write a Control Script, for example:

```
var headerObj = {
       "category": ["invoices"]
};
merge.context.sections["Content"].addHeader("X-SMTPAPI",
JSON.stringify(headerObj));
```

Note

Sendgrid strips out their mail headers. The results need to be verified via their Dashboards (e.g. the Stats section lets you verify the stats for specific categories). Alternatively one can use their Web API to retrieve stats in JSON format. To view the category stats, log in to Sendgrid and choose: Stats > Category Stats > your category name.



MailGun

Dashboard: https://mailgun.com/cp/stats

Documentation: https://documentation.mailgun.com/api-sending.html#sending

Add a *name* meta tag with the value *X-Mailgun-Tag* and set its content (for example: *invoices*). You can make this setting in the **Email section properties** (see "Meta information" on page 186).
Alternatively you could write a Control Script, for example:

```
merge.context.sections["Content"].addHeader("X-Mailgun-Tag",
"invoices");
```

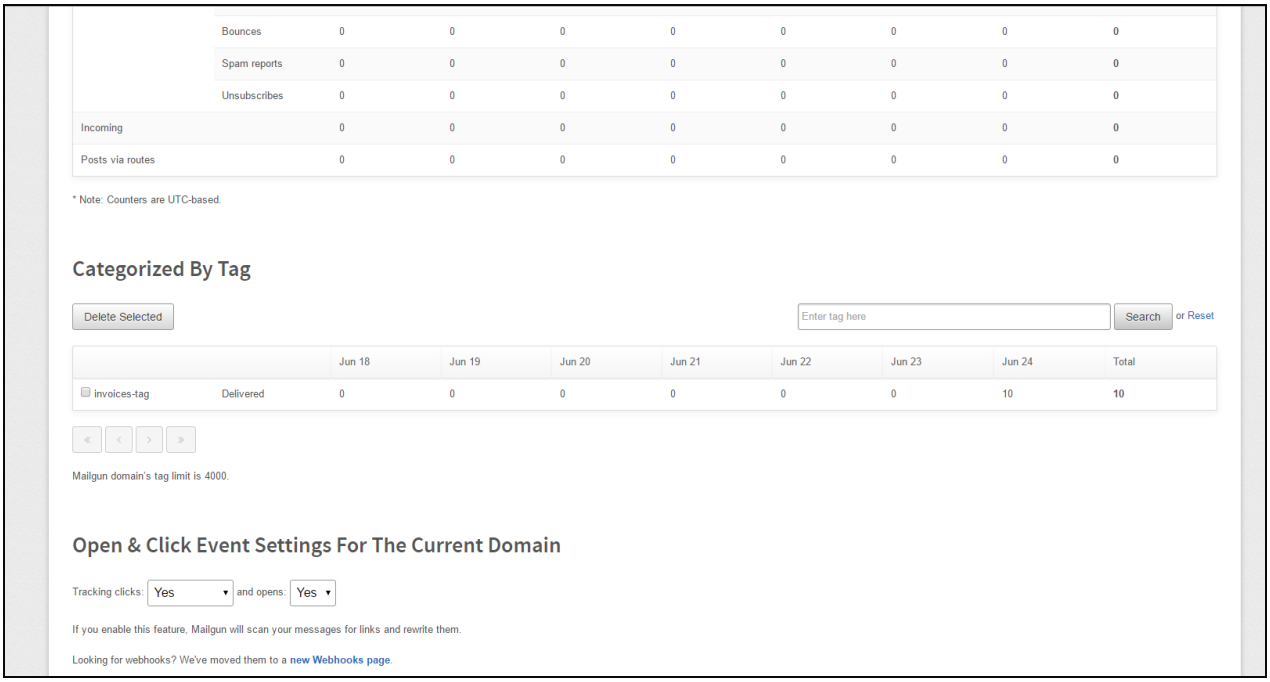
You may specify multiple categories, as follows:

```
var headerObj = {
"category": [
```

```
"dogs",
"animals",
"pets",
"mammals"
]
};
var section = merge.context.sections["Content"];
section.addHeader("X-SMTPAPI", JSON.stringify(headerObj));
```

Note

The Mailgun tag allows you to view the stats per tag. Mailgun has a quick refresh and stats are available almost instantly.



MailJet

Dashboard: https://app.mailjet.com/dashboard

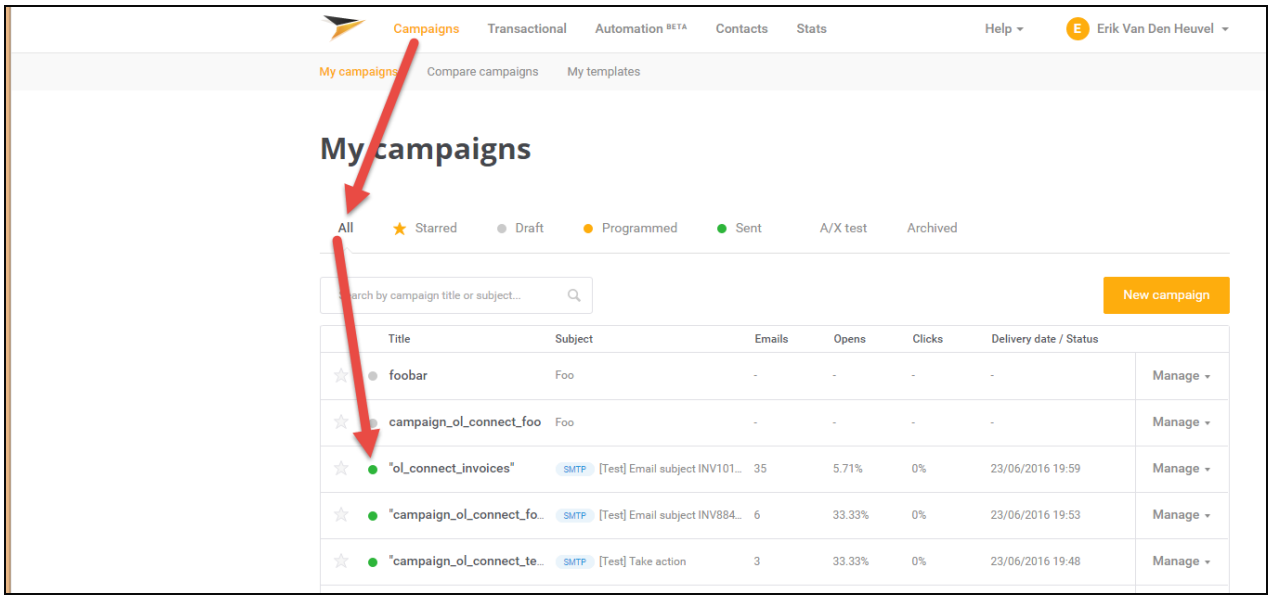
Documentation: https://app.mailjet.com/docs/emails_headers

To assign a campaign, set a meta tag of the type *name* with the value *X-Mailjet-Campaign* and set its content to the name of the campaign (for example: *invoices*).
You can make this setting in the **Email section properties** (see "Meta information" on page 186).
Alternatively you could write a Control Script, for example:

```
merge.context.sections["Content"].addHeader("X-Mailjet-Campaign",
"invoices");
```

Note

Mailjet strips out their own mailheaders like X-Mailjet-Campaign. The results can only be verified via the respective campaign stats page in the Mailjet dashboard. There is no need to pre-create the campaign: adding it to the email header via a Control Script auto-generates the campaign. To view the campaign, login to Mailjet and choose: Campaigns > All.



Optimizing a template

This topic describes some ways to optimize a template in order to speed up the output process. However, the template itself is not the only factor to be taken into account. The server

configuration and hardware configuration have an influence on the output speed as well. For advice on server configuration and hardware configuration, see "Performance Considerations" on page 23.

Scripts

In the process of output generation, the execution of scripts may take up more time than necessary. Here's what you can do to shorten that time.

- **Use efficient selectors**. Using very precise selectors in scripts will be much faster than using a text selector, especially on large documents. See "Use an ID or class as selector" on page 409.
- **Optimize your scripts**. Custom scripts with non-optimized loops and unnecessary DOM manipulations can slow down Content Creation. Use the Designer's test facilities to find out which scripts can be improved (see "Testing scripts" on page 403 and "Optimizing scripts" on page 408).
- **Only run the necessary scripts**. Normally the Designer will run all scripts for each and every record and section. You can save time in the process of Content Creation by organizing scripts in folders and setting their execution scope or even disabling them (see "Managing scripts" on page 400). Note that loading a JavaScript library is generally very fast and is only done once for the record set.
- **Use a fast network and internet connection** or avoid loading external or internet resources. Using images, JavaScript or CSS resources located on a slow network or on a slow internet connection will obviously lead to a loss of speed. While we do our best for caching, a document with 5,000 records which queries a page that takes 1 second to return a different image each time will, naturally, slow output generation down by up to 83 minutes.
- Make sure to use **optimized graphic resources**. For instance, avoid using images with transparency where no transparency is needed.

Images

- Make sure to use **optimized graphic resources**. For instance, avoid using images with transparency where no transparency is needed.
- When a template that contains lots of images is merged with a large record set, the many file requests may slow down the process of output generation. One solution is to combine the images into a single image file (an 'image sprite') and display the part that holds the

image. This reduces the number of file requests and can improve the output speed significantly.

Creating and using image sprites

Step 1. Create a file that contains a collection of images.

Static images may go in any type of image file. Store images that need be added dynamically to the template, in one PDF file, one image per page.

There are several tools to combine image files into a singe PDF. **ImageMagick** is one of them. You could use the convert command of the ImageMagick library:

```
convert C:/myimages/*.jpg C:/myimages/image-collection.pdf
```

You could also use **Connect Designer** itself: create a print template with the size of your images and set the page margins to 0. Create a script that loops over your images and adds them to the text flow of the template. Subsequently generate PDF output and use the resulting file as your collection file.

Step 2. Add the file that contains the collection of images to the template's Resources (see "Adding images" on page 273).

Step 3. Display part of the collection file as an image in the template.

- **Static images** that are part of an image file can be displayed via Cascading Style Sheets (CSS). This technique is much used in web design. In this technique, the file that contains a collection of images is called an **image sprite**. The trick is to create a Box (or Div) for each image and give that box an ID (see "Boxes" on page 256). Then use the ID in a style sheet to select the Box and write a style rule (see "Styling templates with CSS files" on page 296) that sets its background image to the image sprite and positions the image. For an explanation and examples of this style rule, see https://www.w3schools.com/css/css_image_sprites.asp.
- **Dynamically added images** are loaded in a script. To retrieve one page from a PDF file in a script, add the page parameter to the file path and set that as the source of the image. Here is an example (assuming that the page number is stored in a variable `pageNumber`):

```
var imageStr = "";
var imagePath = "file:///C:/image-collection.pdf?page=" +
pageNumber;
imageStr += '<img src="' + imagePath + '">';
results.after(imageStr);
```

Print Manager

The **Print Manager** provides a means of managing print jobs in a print production environment. It provides the ability to queue jobs, to set hot-folders to allow jobs to run the instant they arrive in a live hot-folder, to re-print jobs, to preview job output, to halt job printing, to organise or re-organise the sequence of production printing, to monitor the current state of printers, to restart paused printers, and such like.

The **Print Manager** is primarily designed to run in conjunction with IPDS output, as it *must* be used to send IPDS output for IPDS printers. It can also be used in conjunction with PCL and PostScript output, but with less options available.

IPDS printing requires a *print stream* and PrintShop Mail Connect only produce *files*, thus the intermediary **Print Manager** is *required* to bridge the communication divide to IPDS printers. The **Print Manager** feeds IPDS output directly to the printer as a print stream, rather than as a file.

Neither PCL nor PostScript printers require a *print stream*, so both those output types can be sent directly to a printer from within Connect. However, you could opt to generate spool files from Connect, and then use the **Print Manager** application to provide some extra print management functionality, such as organizing job priorities, printer queues and job reprints.

The **Print Manager** can be installed standalone and can be installed on more than one machine.

Print Manager Introduction Video

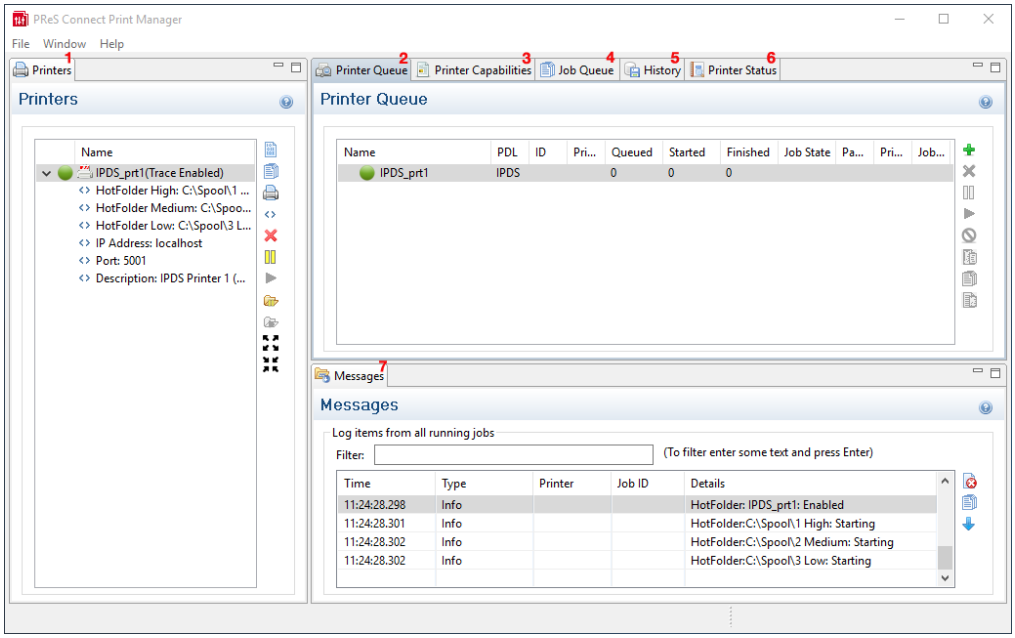
Print Manager usage

For a full description of how to use **Print Manager**, see the "Print Manager Interface" on the facing page

Print Manager Interface

The Connect **Print Manager** Interface is broken up into multiple tabs. These can be re-organised and moved around and re-sized as desired. Double mouse clicking on a tab will make the tab expand full size, to take up the entirety of the application window, or contract the tab back to original size, if already expanded.

The **Print Manager** display initially appears as in the following image (and can be re-set to this view at any time, via the **Windows > Reset Perspective** menu option):



1. "Printers pane" on the next page
2. "Printer Queue pane" on page 1038
3. "Printer Capabilities pane" on page 1052
4. "Job Queue pane" on page 1047
5. "Job History pane" on page 1050
6. "Printer Status pane" on page 1052
7. "Messages pane" on page 1049

Menu options

- **Windows > Preferences ...**: A variety of **Print Manager** settings can be changed through this menu option. Information about the available preferences can be found here: "Print Manager Preferences" on page 1054.
- **Windows > Reset Perspective**: Reset the **Print Manager** perspective. This returns all the **Print Manager** panes (which can be moved around or re-sized as desired) to their original default positions and sizes.

Printers pane

The Printers pane allows you to add or remove printers to the **Print Manager**. Once a printer is added here it will be available for use within the "Printer Queue pane" on page 1038.

The following information (all editable via the "Printer Settings Dialog" on page 1033) is displayed for each printer:

- **Printer Name**: Displays the name of the printer.
  The background color and the icon to the left of the printer indicates the current status of the queue. The status options are:
  - **Green**: The queue is active and any job sent to it is immediately streamed to the printer.
  - **Yellow**: The queue is paused and jobs will wait before being streamed to the printer.
  - **Red**: The printer is stopped due to a physical error (such as running out of toner) on the printer.
- **<> HotFolder** : Displays the paths of any hot folders that are set.
- **<> IP Address**: Displays either the IP Address or machine name of the IPDS printer.
- **<> Port (IPDS Only)**: Displays the port the printer communicates on.
- **<> Queue (PCL and PostScript Only)**: Displays the printing file type.
- **<> Description**: Displays printer description, if one exists for the printer..

The following controls are available to the right of the printer list, as well as via right-mouse click context menus:

- **Add job to selected printer**: Click to add file(s) to the printer queue for the selected printer. These jobs will then appear in the queue for that printer in the "Printer Queue pane" on page 1038.

- **Add job with Page Range**: Click to load a file directly to the printer queue while specifying a page range to print.
- **Add Printer**: Click to add a new printer using the "Printer Settings Dialog" on the next page.
- **Edit Printer**: Click to edit the currently selected printer using the "Printer Settings Dialog" on the next page.
- **Import Printer**: Click to import a Printer definition from a previously saved external file.
- **Export Printer**: Click to save current Printer definition to an external file.
- **Delete**: Click to delete the currently selected printer.
- **Stop Printer Queue**: Click to stop the currently selected printer(s) queue, which prevents any new jobs from being printed on the stopped printer(s). The printer status icon in the "Printer Queue pane" on page 1038 turns yellow, indicating that the printer is not active.
- **Start Printer Queue**: Click to resume the currently selected printer(s) queue. Any job within the queue assigned to the printer(s) will then start being printed. The printer status icon in the "Printer Queue pane" on page 1038 will turn green, to indicate that the printer is active.
- **Pause HotFolder**: Click to pause the hot folder for the currently selected printer. This does not prevent jobs in the printer queue from generating output - only new jobs will stop being captured in the Hot Folder. This button is only activated when the Hot Folder is currently active.
- **Resume HotFolder**: Click to resume capturing new jobs in the Hot Folder for the currently selected printer. This button is only available when the Hot Folder is paused.
- **Expand**: Click to expand the current printer details within the Printer pane, if not already expanded.
- **Collapse**: Click to hide the current printer details within the Printer pane, if they were currently expanded.
- **Disable Printer**: Click to disable the selected printer(s). This either removes the selected printer(s) from view in the Printer Pane (if the **Hide disabled printers** option is active in the General tab of the Preferences) or highlights the printer(s) in red, to visually indicate that they are disabled.

> **Note**
>
> If the **Hide disabled printers** option is active in the General tab of the Preferences, any disabled printers will disappear from the Printers Pane.
> To return them to view, the **Hide disabled printers** option must be de-selected in the Preferences.

- **Enable Printer**: Click to re-enable the selected disabled printer(s).

Printer Settings Dialog

This dialog allows you to add printers to the **Print Manager**, or to modify settings for those already added. The settings differ between printer types, with IPDS printers having more options than other printers.

Each logical grouping of printer setting appears in its own subsection within the dialog. These can be folded up or down (to make them visible, or hidden) by clicking on the blue title bar atop each subsection.

Once the settings are complete, press the Finish button to apply the new settings.

In the case of adding a printer (via the **Add Printer** option), this adds the newly entered printer to the **Print Manager** and automatically makes it available in the "Printers pane" on page 1031 and the "Printer Queue pane" on page 1038.

If modifying an existing printer (via the **Edit Printer** option) the printer settings are then updated throughout the **Print Manager**.

To test the printer settings, one can run **Check Printer Capabilities** in the "Printer Capabilities pane" on page 1052. The printer options that can be set include:

Printer Details

- **Enable Printer**: Check to enable the printer settings.
  Printers can be Enabled or Disabled using this check box, or via the Enable / Disable options in the "Printers pane" on page 1031.
  - **Printer Type**: Select the type of printer to add, from the drop down list.
  - **Name**: Enter a name that describes the printer in the list.

- **IP Address**: Enter the IP Address or machine name of the printer.
- **Port (IPDS Only)**: Enter the port on which the printer communicates.
- **LPR (PCL and PostScript Only)**: Enter the printer queue.

> **Tip**
>
> RAW will work as a generic queue option for most PCL and PostScript printers.
>
> Those PCL and PostScript printers that support multiple queues can have multiple entries in the Print Manager. One for each Queue.

- **Hot Folder** subsection: Defines folders from which print output files are *automatically* captured and sent to the appropriate printer queue.
  - **Enable Hot Folders**: Check to enable hot folders. Browse to set the folders via the Browse ( ) button.

    Up to three Hot Folders can be set. One each for **High**, **Medium** and **Low** priority queues.
    An additional "**Set Job on Hold**" option is available which assigns jobs to the target printer but puts them 'On Hold'. They will not be taken from the queue and processed until manually released.

- **IPDS Options** subsection: IPDS specific options contained in 3 tabs. The tabs are:
  - **Extended Options** tab:
    Defines logging of printer activities. The options are:
    - **Page Counters:** IPDS Printers return counters for various categories, from which you can chose.

> **Note**
>
> The Print Manager will use whichever is selected in this list box to determine when a job is Finished. Once the Counter for this selection equals the total number of pages in the job, the job will be considered Finished and it will display as such within the

Printer Queue pane" on page 1038.

What choice you make here is dependent upon the capabilities of the printer. Not all printers will provide all options. For example, some printers will only return when the job is Committed, rather than when it has been Received or Stacked.
The full range of choices available to the Print Manager (but not necessarily the IPDS printer) are:

- **Received**: The amount of pages received by the printer.
- **Stacked**: The amount of pages stacked by the printer.
- **Jam Recovery**: If a paper jam is encountered, the page number at which the printer jam occurred.
- **Committed**: The number of pages committed to the printer.
- **Operator View**: This counter contains the number of pages that have passed the view of the printer operator. It increments by the number of pages on a sheet when the last copy of the sheet passes the view of the printer operator.

- **Ack Pages:** The number of pages to send to the printer for printing before sending an acknowledgement request.

  Acknowledgement requests slow the printing process down, as the printer needs to process the request and return the information.

  The default Ack Pages number is now 10 pages which means the Print Manager will send an acknowledgement request to the printer in increments of 10 pages, until the end of the job. Given the overhead in processing such requests, the smaller the Page number, the slower the printing becomes, but the higher the granularity of the reporting and error correction. If speed is of the essence you might like to increase the number to 100.

  We recommend retaining the default value, due to the granularity of reporting and the error correction this allows.

- **SNMP:** Select this option to allow IPDS printer(s) and Print Manager to communicate back and forth.

  > **Note**
  >
  > Not all IPDS printers will support SNMP, but most contemporary printers are likely to.

  This option is enabled by default, as it provides the greatest functionality.
- **Extended Options:** Select this to allow access to further Extended Options.
  The Extended Options consist of:
  - **Create Trace File:** Select this to log all communications to a log file.
    The trace file will be sent to the following folder:`C:\Users\[UserName]\Connect\trace\PrintManager`
    The `[UserName]` would be the currently active user's Windows login name.
  - **Legacy Mode**: As of Connect 2021.1 outputting *Multipage PDF containers* is the default for IPDS printers.
    To disable Multipage PDF containers, select the **Legacy Mode** option.

    > **Note**
    >
    > When printing on high volume IPDS printers, it is best if the printer does not pause often. On continuous feed printers in particular, pauses can lead to malfunctions as well as cause delays. It's not just the pause itself, but also the additional time spent whilst the printer slows down and then speeds up again after the pause. The faster the printer, the bigger an issue this becomes.
    >
    > Most modern IPDS printers support Multipage PDF

containers, which can be used in such a way to as remove pauses between print jobs.

Outputting jobs using Multipage PDF containers means high-end printers can be largely prevented from pausing. So Upland OL recommends using **Legacy Mode** only if your printer does not support Multipage PDF containers, or you run into issues.

- **Twin Printer Mode**: Select this mode when using a twin printer.

- **End of Job Preferences (IPDS only)** tab:
  Select from these options to determine job end processing. More than one option can be selected simultaneously.
  - **Edge Marks:** Select this to add an Edge Mark to the final page of a job. This is commonly used to indicate job separation in Continuous Form printing.
  - **Stack Received Pages:** Add blank pages to the end of the job in order to push the completed job to the stacker.
  - **Print Buffered Pages:** Print any/all buffered Pages. This selection causes the printer to schedule all buffered data for printing prior to sending an Acknowledge Reply.
    Buffered data is page data not reflected by the committed page and copy counters.
  - **Eject to Front Facing:** For cut-sheet media, this selection causes the next received page to be printed as the first page of the next sheet.
- **Error Handling Preferences** tab:
  Select these to **STOP** printing upon encountering any of the following errors (more than one option can be selected):
  - **Undefined Character Check:** If an undefined character is encountered in the job. For example, the Euro symbol might be present in the job, but missing from the printer.
  - **Report Page Position Check:** When printed beyond the edge of the page boundaries. For example, an image that extends beyond the edge of the page.

- **Handle Errors:** Select for how to handle any/all other optional errors. The choices are:
  - **None** - ignore all other errors.
  - **Auto** - allow default error handling.
  - **Prompt** - prompt for operator action on error.

Description

An optional printer description can be entered here, if desired.

Buttons

- **Finish** button: Click to apply the selected settings to the printer. This applies to both new printers and to update existing printers.
  New printers will appear in the "Printers pane" on page 1031 and the "Job Queue pane" on page 1047.
- **Cancel** button: Click to cancel the selection of printer settings. This exits the dialog, without any changes being made

Printer Queue pane

The Printer Queue pane displays all the printers available for printing (as set through the "Printers pane" on page 1031), their status and the job queues associated with each printer.

This pane allows you to add or remove print jobs to the printer's queue; to pause and then restart printing (at either the individual job level, or at the printer level); to reprint jobs: to print or reprint selected page ranges and other common print production tasks.

This pane would be the main screen used by print operators to control production printing.

> **Note**
>
> Not all of the functionality on this pane will be available to all printers. Much of the Printer Queue functionality has been specifically designed to work with IPDS printers and is

currently only available for IPDS printers and output files.
Where such restrictions apply will be noted on this page.

Job Status

When a print job is being processed by the **Print Manager** the job is renamed to reflect the current processing step. A new file extension is added to the job, and updated in real time, as described in the following table:

filename.ipds	A print job that is in the queue but which has not yet begun processing.
filename.ipds.**Started**	A print job that is now active. It is currently being processed.
filename.ipds.**Committed**	The print job has successfully been sent to the printer and is now being Committed.
filename.ipds.**Finished**	This print job has been processed.
filename.ipds.**Queued**	This print job has been resubmitted and is queued for processing.

These file extension entries generally match those found in the Printer Queue Table "*Job State*" field.

Printer Queue Table

This table displays information about both the available printers (which can be added or removed via the "Printers pane" on page 1031) and their associated job queues.

Clicking on selected column titles (such as *Name, PDL* and *Priority* columns) will re-sort the Queue display based upon the column selected. Clicking on the column title a second time will reverse the order.

The fields the table contains are broken down as follows:

- **Name**: Displays the name of the printer and beneath it are the names of all the print jobs currently residing in the printer's job queue.

  Print jobs can be added or removed from the printer's queue by a variety of means. Such as dragging and dropping jobs between the "Job Queue pane" on page 1047 and the

Printer Queue panes. When a job is added from the Job Queue pane thus, it is then removed from the Job Queue pane thereafter.

Jobs can also be added by right-click context menu options available at the printer level. The options include adding complete jobs via **Add Job**, or partial jobs (IPDS only) via the **Add Job with Page Range** option.

Jobs can also be dragged from one printer to another within the Printer Queue pane itself.

- **PDL**: Displays the printer language (for both the printer and print jobs).
- **ID**: An unique identity number that is automatically generated for every job.
  The first few letters of the ID are coded to indicate how the job was added to the queue. The relevant codes are as follows:
  - **HTF**: Hot Folder capture. Any job that lands in the printers **Hot Folder** is automatically added to the Printer Queue (at the appropriate priority) and given this code.
  - **DRP**: Jobs "dragged and dropped" directly into the Print Manager (either from Windows Explorer, or from the "Job Queue pane" on page 1047) are given this code.
  - **BRW**: Code for jobs added to the queue via the **Add Job** file browse option.
  - **RSB**: Code used for jobs that have been **Resubmitted** to the printer.
- **Priority**: Displays the priority with which the job is to be printed: Low, Medium or High. Jobs are processed in the queue in the following order: High, then Medium, then Low. Job priorities can be changed on the fly, through the **Set to xxxx Priority** context menu options.
- **Queued**: At the Printer level this displays the number of jobs currently queued to the printer. At the print job level it displays the date and time that the job was added to the queue.
  This counter updates in real time.
- **Started**: At the Printer level this displays the total number of jobs that were started on this printer. At the print job level it displays the time an individual job started printing.
  Please note, this counter updates in real time, but always adds to the existing Printer total. Use the **Reset Printer Counters** option to re-set the current Printer total to zero.
- **Finished**: At the Printer level this displays the total number of jobs that have finished on this printer. At the print job level it displays the time an individual job finished printing.

Please note, this counter updates in real time, but always adds to the existing Printer total. Use the **Reset Printer Counters** option to re-set the current Printer total to zero.

> **Note**
>
> For IPDS printers the "*Page Counters*" entry in the "*Extended Options*" section of the "Printer Settings Dialog" on page 1033 sets the criteria that determines whether a job is Finished or not.
>
> For example, if the "*Page Counters*" entry is set to Stacked, then the job is only considered Finished when the printer's Stacked pages counter equals the amount of pages in the job.

- **Job State**: Displays the individual print job status. These could be Queued, Finished, Started, and such like.

> **Note**
>
> For IPDS printers the "*Page Counters*" entry in the "*Extended Options*" section of the "Printer Settings Dialog" on page 1033 sets the criteria that determines whether a job is Finished or not.
>
> For example, if the "*Page Counters*" entry is set to Stacked, then the job is only considered Finished when the printer's Stacked pages counter equals the amount of pages in the job.

- **Page Range**: Displays the range of pages to be printed. The default is to print the entire page range, but this can be cut down to a specific sub-set of pages via the **Resubmit Range of Pages** option.
- **Printed Pages**: At the print job level this displays the number of printed pages in the current job. At the Printer level this displays total number of pages printed on the printer. Please note, this counter updates in real time, but always adds to the existing Printer total. Use the **Reset Printer Counters** option to re-set the current Printer total to zero.
- **Job Pages**: Displays the total number of pages in the job.

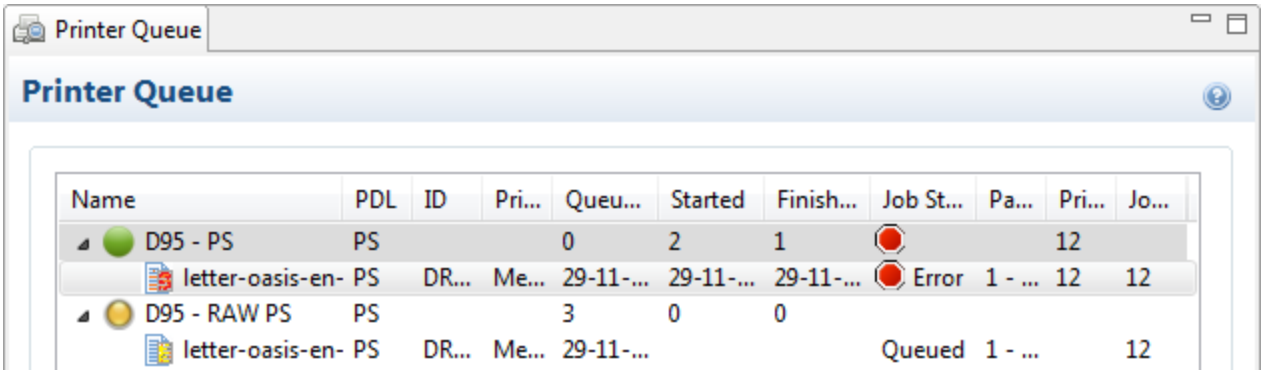
The following options are available via right-mouse click context menus. The options that display will be dependent upon what has been selected (Printer or Job) and the current state of the selected item.

Printer Queue Options

> **Note**
>
> These options will only be made selectable when they are actually currently available. If an option is not available to the current selection or setting, it will not be accessible.

- **Start Printer Queue**: Select to start a printer which has been stopped, or which has been recently added.
  Stopped printers have a yellow status icon to indicate that the printer is not currently active. Started printers have a green status icon, to indicate that the print queue is currently active.

  The PostScript printer entry in the following image shows how a printer error state is represented within Print Manager when a printing or connection error occurs. The printer is still active (and thus the printer status icon remains green), whilst the job itself is stopped due to the error. This printer is, in effect, in a paused state whilst it waits upon the error being fixed, or the job being removed from the queue.



- **Stop Printer Queue**: Select to stop an active printer in the printer queue.
  Once selected, that printer's queue is immediately stopped and the printer status icon turns yellow to indicate that the printer is not active.
- **Refresh**: Select to refresh the connection to a printer. This is handy when the communication between printer and Print Manager has encountered issues.

- **Add Job**: Select to add new job(s) to the selected printer. Several jobs can be added at once.
- **Add Job with Page Range** Select to add a specified sub-set of an IPDS job. This launches the "Add Job with Page Range Dialog" on page 1045 in which the job and it's required page range can be set.
- **Remove Finished Jobs**: Select to remove any Finished Jobs from this printers Printer Queue display, making the visual monitoring of active and scheduled jobs easier.
- **Reset Printer Counters**: Select to update the printer information job counters. This resets the number of jobs Queued, Started or Finished.
- **Stack Received Pages**: Select to push out the last page committed into the printer stacker.

Printer Queue Job Options

> **Note**
>
> These options will only be made selectable when they are actually currently available.
> If an option is not available to the current selection or setting, it will not be accessible.

- **Preview** button / context menu option: This launches a "Preview pane" on page 1046 that displays what the print job looks like.
- **Pause Job**: Select to temporarily pause the current IPDS job on the printer queue.
  This option only works for IPDS print jobs and the option is only available when the printer queue is active.
  Once a job is paused, the job queue for this printer is in effect paused as well, until the paused job is either restarted or cancelled.
- **Resume Job**: Select to resume a currently paused IPDS job. This option is only ever active when an IPDS print job is paused in the printer job queue.
- **Cancel Job**: Select to cancel the currently active job (even if paused). This option only works for IPDS print jobs and the option is only available when the printer queue is active.
  Cancelling the active job will also stop the printer, rather than have it continue onto the next job in the print queue.
  A cancelled job can be restated at a late time. It can either print in full, via **Resubmit**, or

continue from where the job was cancelled, via **Resubmit Range of Pages** and specifying the starting page as the page the job was cancelled on,

- **Refresh**: Select to refresh the job information. This allows close monitoring of individual jobs.
- **Remove Job**: Select to delete the currently selected job(s) from the Printer Queue.
- **Resubmit Job**: Select to restart the selected job(s) from the beginning. If the original job had been processed using a page range selection, it will be resubmitted using the same page range.
- **Resubmit Range of Pages**: Select to resubmit an IPDS job, but with a specified page range selection. This launches the "Add Job with Page Range Dialog" on the next page in which the job and it's required page range can be set.
  This option is very useful when recovering from a print job failure, when a print job has only been partially printed, and you wish to complete the job. The dialog box will present you with a range of options as to where to re-start the job from.
- **Re-assign to Job Queue**: Select to take the selected job(s) out of the Printer Queue and place it back in the generic Job Queue. The same can be achieved by dragging and dropping the job from the Printer Queue to the Job Queue.
  This option is available when the print Job has yet to be processed. Either when the Printer is in a paused state, or if the Printer is active but has not yet reached this Job in the Printer Queue.
- **Set Job on Hold**: Select to hold this job on the printer queue. When held a job will not print, until it is once again activated by assigning a new job Priority. At which point it becomes active again, and will be process at the selected priority.
- **Set to Low Priority**: Select this to set the job priority to Low. This will immediately drop the job below any/all Medium or High priority jobs in the queue, as well as taking a job off Hold, if it had been held.
- **Set to Medium Priority**: Select this to set the job priority to Medium. This will immediately drop the job below any/all High priority jobs in the queue, but above all Low priority jobs, as well as taking a job off Hold, if it had been held.
- **Set to High Priority**: Select this to set the job priority as High. This will immediately push the job above any/all Medium or Low priority jobs in the queue, as well as taking a job off Hold, if it had been held.
- **Directory**: Select to open a Windows Explorer window set to the directory containing the job file.

- **Set Finished**: Select this to change a Committed job status to Finished. This option is only available to Committed jobs. It is a way to manually set a completed job to Finished, when the Print Manager has not automatically done so (as can happen, in some scenarios).

Add Job with Page Range Dialog

The **Add Job with Page Range** dialog box works with IPDS output only. It allows the selection of a page range to print, from IPDS spool files.

It can be used to print a section from within the middle of the file, or it could be used to select a starting point and then print all the pages thereafter.

Job Name details section:

- **File Name**: Paste the print job filename and folder into the entry box, or use the Browse button to browse for the print job.
- **Start Page**: Select the page number at which to start printing.
- **Last Page**: Select the ending page number.
  This defaults to the last page in the print job, but can be set to any page that follows the **Start Page** selection.
- Preview **Preview** button : Use this button to launch the "Preview pane" on the facing page.
  The preview will display the current job, starting at the page selected as the **Start Page** in the page range.
- **Priority**: Select the job printing priority.

Select Start Page details section: ***This section only appears when resubmitting a previously processed IPDS job.*** It allows you to set the starting page based upon the following criteria:

- **Start Page**: The first page of the job. If the job was selected with a page range, this would be set to the beginning of that previously selected page range. Otherwise it will always be page 1.
- **Pages Committed**: Select this to start from the last page Committed to the Printer.
- **Page Jammed Recovery**: Select this to start from the Paper Jam Recovery page counter entry.
- **Pages Received**: Select this to start from the Pages Received counter entry.

- **Operator View**: Select this to start from the Operator View counter entry.
- **Pages Stacked**: Select this to start from the Pages Stacked counter entry.

> **Tip**
>
> When an issue arises with the printer and you wish to restart the job from a recovery point, it would be best to first "Preview pane" below the page at the selected page counter, to confirm that the selected page counter entry is accurate.
>
> In reality, you might need to move the starting page entry backwards or forwards some few pages in order to cater for pages which have been printed, but destroyed in the process.

Preview pane

The Preview pane displays previews of print jobs from with the "Printer Queue pane" on page 1038, the "Job Queue pane" on the next page, the "Job History pane" on page 1050 and the "Add Job with Page Range Dialog" on the previous page.

> **Note**
>
> The Preview will work "out of the box" for both PCL files and standard IPDS files but not for PostScript files, nor continuous IPDS output created as Multipage PDF containers. Previewing either of these types of files requires the **Ghostscript** software be installed first.
>
> Ghostscript is a free industry standard PostScript interpreter/renderer that can be downloaded from here: https://www.ghostscript.com/download/
>
> Select the **Ghostscript AGPL Release** version that best matches your Operating System, then download that installer and run. PostScript and continuous IPDS output Previews will then be available, the next time you launch the Print Manager.

The Preview pane allows you to view all the pages in a print job, via the Preview toolbar:

You can step through print jobs using the page step buttons, or move straight to a specific page by entering a number in the page number box. You can also customize the preview viewing settings as you wish.

> **Note**
>
> Print jobs featuring an associated Connect "Output Creation Presets" on page 1001 which have "Imposition options" on page 774 set to Landscape will not **Preview** in Landscape, but rather in Portrait mode. The printout is unaffected, however.

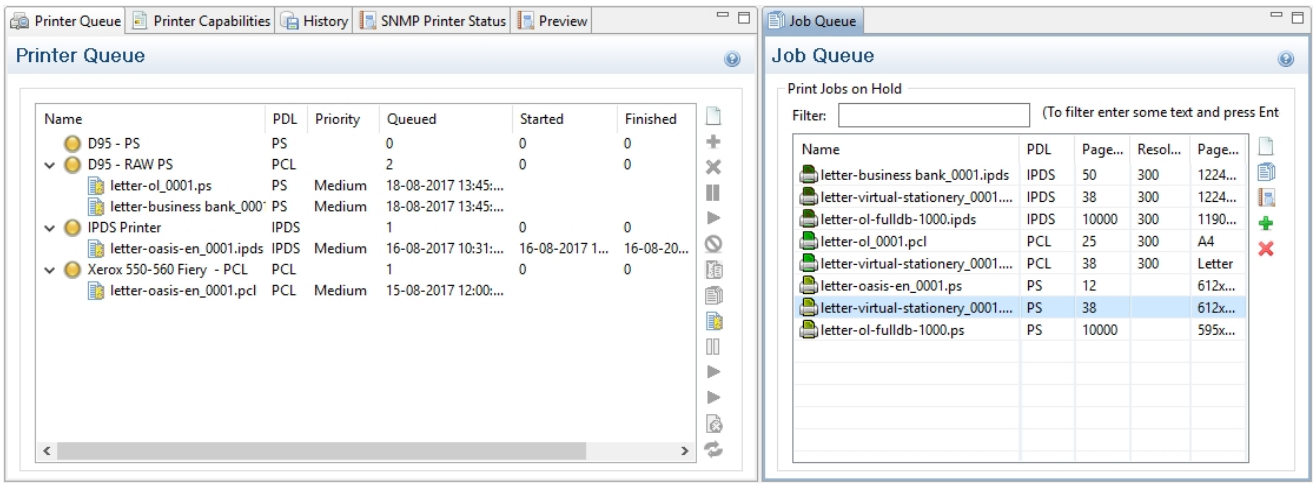
Job Queue pane

The Job Queue pane is a centralized location designed to allow the consolidation and management of print Jobs.

Jobs are not printed from here. The Job Queue could best be considered a staging area, where one can consolidate jobs in preparation for production printing.

You can add Jobs to the queue by "dragging and dropping" them from a Windows Explorer window, or from the "Printer Queue pane" on page 1038, or by using the **Add Job** option.

Once in the Job queue, jobs can be previewed to confirm the contents are correct, and selected for printing by "dragging and dropping" the print Job into the "Printer Queue pane" on page 1038.
To do this, you must have both screens visible at once, as seen here:



Note

Print Jobs ***aren't actually scheduled for printing*** in the Job Queue. That is done in the "Printer Queue pane" on page 1038. This Job Queue is a job holding/staging queue, rather than a production printing queue.

- **Filter**: Use this search functionality to reduce the data-set displayed in the table.
  Type in a search word and then hit Enter to filter the table using the entered search word. For example, select "business" to display only those entries that feature the text "business".

  To clear the filtering, delete the contents of the Filter box and press Enter.

Job Queue Table

- **Name**: The name of the job filename, as captured or added to the Print Manager.
- **ID**: An unique identity number that is automatically generated for every job.
  The first few letters of the ID are coded to indicate how the job was added to the queue. The relevant codes are as follows:
  - **HTF**: Hot Folder capture. Any job that lands in the printers **Hot Folder** is automatically added to the Printer Queue (at the appropriate priority) and given this code.
  - **DRP**: Jobs "dragged and dropped" directly into the Print Manager (either from Windows Explorer, or from the "Job Queue pane" on the previous page) are given this code.
  - **BRW**: Code for jobs added to the queue via the **Add Job** file browse option.
  - **RSB**: Code used for jobs that have been **Resubmitted** to the printer.
- **PDL**: Displays the job printer language.
- **Page Count**: Displays the total number of pages in the job.
- **Resolution**: Displays the resolution, in DPI, of the job.
- **Page Size**: Displays the dimensions of the job.

To the right of the table and also via right-mouse click context menu within the table, the following options are available:

- **Preview** button / context menu option: This launches a "Preview pane" on page 1046 that displays what the print job looks like.
- **Copy selected items to clipboard** button / **Copy** context menu option: This copies job information (job name and folder path) to the clipboard.
- **Properties** button / context menu option: This launches a job Details Window that displays detailed information about the makeup of the print job. Once the Details Window is open it remains atop the Print Manager thereafter, to allow simple comparisons between jobs.
  To close the Details Window, put the focus upon the Details Window by clicking anywhere within it, and then press the Escape key ("Esc").
- **Add Job** button / context menu option: Click this to add a print Job to the Job Queue.
- **Remove Job** button / context menu option: Click this to remove a print Job from the Job Queue.
- **Refresh** button / context menu option: Refreshes the job(s) to update current page count and the like.
- **Directory** button / context menu option: Opens the folder containing the selected file.

> **Note**
>
> If multiple files are selected, only the folder of the first file selected will be opened.

Messages pane

The Messages pane displays any notice, warning or error message that is generated by the **Print Manager**. This may include indications that jobs have finished or started, or that errors have occurred.

The level of logging can be set in the Print Manager Preferences screen, under General preferences.

- **Filter**: Use this search functionality to reduce the data-set displayed in the table.
  Type in a search word and then hit Enter to filter the table using the entered search word. For example, select "business" to display only those entries that feature the text "business".

  To clear the filtering, delete the contents of the Filter box and press Enter.

- **Messages** table: This contains the message details. Which are:
  - **Time**: The message time-stamp.
  - **Type**: The type of message.
  - **Printer**: The relevant printer, if applicable.
  - **Job ID**: The relevant Job ID, if applicable.
  - **Details**: The actual message itself.

The following controls are available to the right of the Messages table, or via right mouse click context menu:

- **Clear Log Items** button / **Clear Log** context menu option: Click to clear the current log. This does not delete log files, but it does delete the messages displayed onscreen.
- **Copy selected items to clipboard** button / **Copy** context menu option: Click to copy all messages to the clipboard. Useful for pasting messages into external software.
- / **Always select last log entry** button / **Tail** context menu options: Click to toggle auto-scrolling on/off. Auto-scrolling ensures the most recent message is always displayed when it is added to the log.

Job History pane

The Job History pane displays a list of print jobs that have been processed through the **Print Manager** (except for those which have been explicitly removed).

How many print jobs are displayed (and the order of the display) is determined by the **History** settings in the Print Manager "General Settings" on page 1057. The default is for the last 500 jobs to be displayed.

- **Filter**: Use this search functionality to reduce the data-set displayed in the table.
  Type in a search word and then hit Enter to filter the table using the entered search word. For example, select "business" to display only those entries that feature the text "business".

  To clear the filtering, delete the contents of the Filter box and press Enter.

Note

The Filter applies to the entire data set, so if you were to search for the text "Med" the filter would return all those jobs that featured "Med" in the title, plus all those that in any hot folders that featured "Med", plus all jobs whose priority had been set to "Medium".

The search is case insensitive. Thus a search for "med" would return "Medium", as well as "medium" values.

- **History Log**: Displays details on jobs run through the **Print Manager**. The history can be sorted by either processing date or printer, by selecting the column title.
  - **Date**: The date and time when the job was started.
  - **Printer**: The name of the printer the job was printed to.
  - **ID**: The internal ID of the job.
  - **Job**: The name of the print spool file.
  - **Path**: The path from where the job was captured or added.
  - **Start Page**: The page from where the printing started.
  - **Pages**: The pages printed and the total number of pages in the job.
  - **Status**: The final status of the job, such as Cancelled or Finished.
  - **Priority**: The job queue priority the job was printed with.

To the right of the table and also via right-mouse click context menu within the table, the following options are available:

- **Preview** button / context menu option: This launches a "Preview pane" on page 1046 that displays what the print job looks like.
- **Copy selected items to clipboard** button / **Copy** context menu option: Click to copy all messages to the clipboard to paste them in an external software.
- **Always select last log entry** button / **Tail** context menu option: Click to toggle auto-scrolling so that the last message is always displayed when it appears.
- **Remove Job** button / context menu option: Click to delete job from history.

Printer Capabilities pane

The Printer Capabilities pane displays the technical capabilities of any printer that has been added to the "Printers pane" on page 1031.

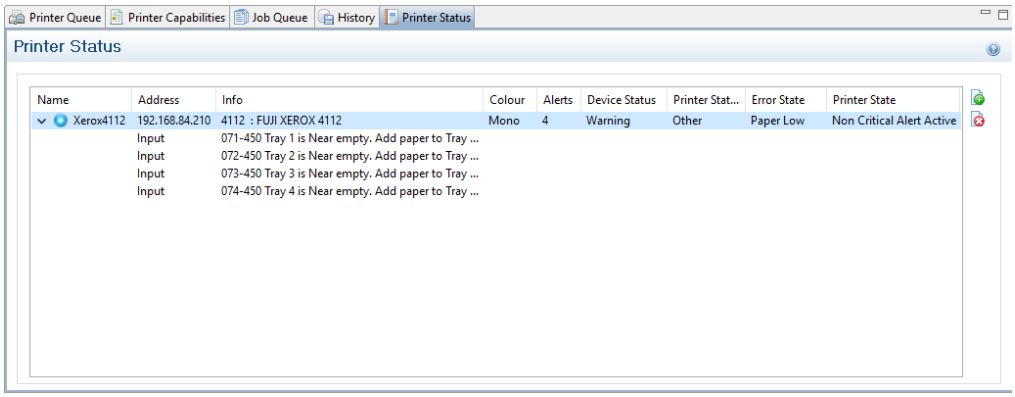
- **Filter**: Type in a filter for a specific capability name and press Enter to filter the results below.
  To clear the filter, delete the contents of Filter edit box, and press Enter.
- **Name**: Displays the list of printers and their capabilities. By default, capabilities are not displayed for performance reason. To display the capabilities, right-click on the printer and select **"Check Printer Capabilities"** or click on the button to the right.

> **Note**
>
> Requesting a printer's capabilities can take some time, and not all printers will support it.
>
> For PostScript and PCL printers you should first set up monitoring of the printer, in the "Printer Status pane" below.

  - **Check Printer Capabilities** button / context menu option: Each printer capability is listed under the Printer, in separate divisions. Capabilities are read directly from the printer.
  - **Export Printer Capabilities** button / context menu option: Export the printer capabilities to JSON file. This contains all the relevant printer capabilities, in a text file format.
  - / **Expand / Collapse** button / context menu option: Expand or contract the printer details.

Printer Status pane

This pane can be used to monitor the state of PostScript and PCL printers. It contains general printer information, such as the printer's B&W or color settings, as well as the quantity of paper in trays, any outstanding alerts or errors and whether the printer is currently in use, or awaiting a job. An example of this can be seen in the following image:

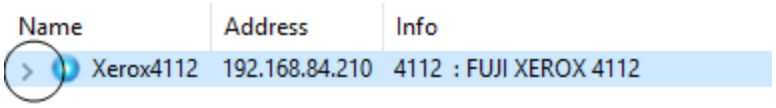


To **Start monitoring a printer** select the start icon to the right of the status display:

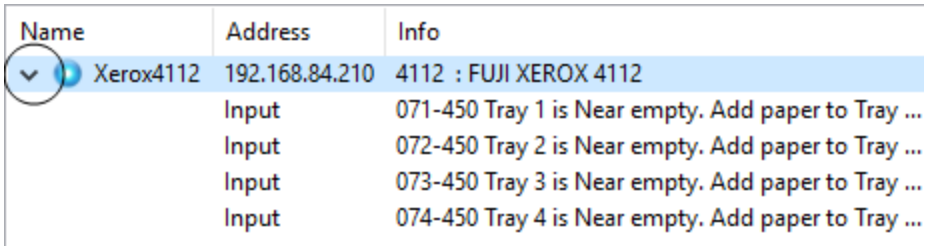
To **Stop monitoring a printer** select the stop icon to the right of the status display:

Once a printer is being monitored, the details can expanded and the individual printer trays displayed. These can be done as follows:

- **To expand the printer details**, press the right arrow besides the printer Name, as seen in the following image.



- **To reduce the printer details**, press the down arrow besides the printer Name, as seen in the following image.



Note

If SNMP monitoring is enabled on a printer, then printer issues will pause the "Job Queue

pane" on page 1047 for that printer.

Print Manager Preferences

The **Print Manager** *Windows* > *Preferences ...* menu option allows adjustment of various **Print Manager** preferences.

There are several groups of preferences which can be changed, or reset to default settings at any time. These groups are:

1. "Database Connection preferences" below
2. "General Settings" on page 1057
3. "Language Setting" on page 1060
4. "Log Settings" on page 1060
5. "SNMP Connection Settings" on page 1062

Database Connection preferences

Dialog used to change the PrintShop Mail Connect back-end Database.
This dialog supports the swapping of the back-end database between various vendor databases. Note, however, that the alternate vendor database(s) *must already be installed and available* in order to swap to them.

This is not a migration tool. It is a simple connection tool, that enables shifting to a different back-end database. Any existing data will **not** be transferred/ migrated between the databases, and any existing File Stores will be cleansed by the Clean-up Service after the swap.

When shifting to a different back-end database, the changes won't be applied until PrintShop Mail Connect is restarted. Including the Connect services. A full machine restart is recommended, as this provides the cleanest restart of all the services.

- **Basic Connection Settings** selections:
  - **Database vendor**: Select the database type from the drop down list.

    > **Note**
    >
    > Moving from one vendor database to another will reset all screen selections to defaults, regardless of what may have been previously selected.

  - **Database URL**: This is a read-only summation of the current database connection settings.

    > **Tip**
    >
    > If the **Test Connection** button shows that the database cannot be successfully connected to using the selected settings, then the contents of this field could be used to try to connect to the database outside of PrintShop Mail Connect. This should help in determining and refining the acceptable connection options.

  - **Hostname**: Enter the IP Address or alias of the server where database resides.
  - **Database Instance Name**: Enter an existing Microsoft SQL Server's instance name.
    This option only applies to existing Microsoft SQL Server instances, and not for MariaDB or MySQL.
  - **Port**: Enter Port number. The defaults are those which the vendors use by default.
  - **Schema**: The individual database schema, within the vendor database.

    > **Note**
    >
    > If a previously non-existent schema were chosen here, then a new schema of that name will be created within the database when the back-end database swap is applied. The tables within that schema, though, will not be created

until Connect is restarted.

- **Username**: Enter the database login username.

> **Tip**
>
> It is considered best practice for this user to have root privileges.

- **Password**: Enter the password associated with selected username.

- **Advanced Connection Settings** selections:
  - **Maximum concurrent threads**: This option sets the maximum database threads. The maximum setting is determined by the specific capabilities of the machine Connect is installed upon (CPU speed and the amount of cores being the major determinants).

> **Tip**
>
> Leaving this value set to the default maximum *should* be the best option in most circumstances.
> We recommended this entry be left at the default value.

  - **Custom database parameters** table: These are extra parameters which are appended to the database connection URL. The default values are those which have been determined to be useful in connecting to specific vendor databases. For example, in order to use Integrated Authentication with Windows, specify the property `integratedAuthentication` with the value `true`, which adds `integratedAuthentication=true` to the URL.

- **Property**: These are free field text fields.

  > **Note**
  >
  > These fields and their associated values get appended to the JDBC connection and therefore must follow all rules regarding acceptable URL addresses for such.

  - **Value**: The value applied to the associated Property.
  - **Add**: Used to add extra Property values to the table.
  - **Delete**: Used to remove existing Property values from the table.
- **Test Connection**: Use to test if current connection settings will connect to the specified database.
- **Restore Defaults**: Will restore the settings to PrintShop Mail Connect HyperSQL standard defaults.
- **Apply**: When a database connection is confirmed as correct this button becomes active, and is used to actually apply the database swap.

General Settings

- **Hot Folder** group: Define a global hot folder that captures any jobs placed within that folder and places them in the Job Queue pane (which can be considered a holding queue).
  This does not automatically print the jobs, but it does allow them to then be dragged across to the appropriate production printer in the "Printer Queue pane" on page 1038.
  - **Enable Hot Folder** checkbox: Check to enable the global Hot Folder, to have jobs added to the Job Queue pane.
  - **Monitored Folder** entry: This entry is only available if the **Enable Hot Folder** checkbox selected.
    Enter the path for the global Hot Folder, or click the Browse button to browse for the folder.
  - **Add Time Stamp** group: Section used to add a time stamp to any files that arrive in the Hot Folder, in order to ensure file uniqueness.
    - **Add Time Stamp** checkbox: Select to have a time stamp added to any files as they arrive in the Hot Folder.

    - **Format** selection: Select the format of the time stamp to add to files in the Hot Folder.
- **Session Retries**: The number of times Print Manager will try to reconnect to the printer before it stops trying and instead produce a timeout error in the messages.
- **Printer Timeout (sec)**: The startup time (in seconds) allowed to the printer before a timeout error is reported in the messages.
- **Printer Ack Timeout (sec)**: The time (in seconds) that the Print Manager waits for a response (an **Ack**nowledgement) from the printer after sending a response request.
- **Retry Delay**: The time (in seconds) that the Print Manager waits before retrying a job after a timeout.
- **Last print job idle delay**: The time the Print Manager will sit idle after a job, waiting for another job to arrive.
  To avoid having to wait after the last job, you could right-click the printer in the Printer Queue and select "Stack Received Pages" to push out the last page committed into the printer stacker. (See: "Printer Queue Options" on page 1042)
- **IPDS Streamed Reader** checkbox: Select this option if difficulties are experienced with IPDS printers periodically freezing whilst **Print Manager** reads responses from the printer.
- **Log Level**: Use the drop-down to select which Print Manager logging level to use.

> **Note**
>
> This logging relates to Print Manager job processing, and not to internal Print Manger debugging. Settings for the internal logs are catered for in the generic "Log Settings" on page 1060 pages.

The selection of logging chosen here will determine what is displayed in the Print Wizard Message Pane.
Higher logging levels such as "Debug" and "Status" provide more details but will have an impact in terms of processing time and resources required.
You might need to adjust this setting based upon your production facilities, production speeds and logging requirements. Some sites might decide that "Error" logging levels (which only records Errors, and thus should be the minimal logging option) are all they need.

> **Tip**
>
> We recommend leaving the logging level set at the default level of "Info", unless otherwise needed.

- **Printer Preferences** group:
  - **Hide disabled printers**: Select to have any printers disabled in the "Printer Settings Dialog" on page 1033 not appear in the "Printers pane" on page 1031.

    > **Note**
    >
    > Not being able to see a printer that exists in the "Printer Settings Dialog" on page 1033 within the "Printers pane" on page 1031 can be confusing for new users.
    > We would recommend not selecting this option until you are fully comfortable with the Print Manager.

- **Cleanup** group:
  - **Automatically remove finished jobs** checkbox: Select to have jobs in the Print Manager automatically deleted, after a set period of time.
  - **Remove finished jobs after** setting: This option only becomes available if the **Automatically remove finished jobs** checkbox was selected.
    Select the time period (measured in either *Minutes*, *Hours* or *Days*) for which finished jobs are to be retained. When this retention period is completed the job(s) are deleted. The deletion both removes the job from the Printer Queue, as well as deleting the file on disk.
    The job will remain listed in the job history, but can no longer be viewed.

- **History** group:
  - **Limit History view at startup to** selection: Use this to limit the number of jobs that will be loaded in the Job History table. The Job History table becoming very large was found to have adverse impacts upon Print Manager performance. This option allows you to limit the Job History to more manageable levels.
    When limiting the number of jobs shown, it is the newer jobs that are retained.
  - Select how the Job History table should be ordered. Either **Descending** or **Ascending**..

Language Setting

- **Display language**: Select a language from the drop-down list to be used as the language of the User Interface (after the software is restarted).
- **Default Locale**: The default locale sets the locale for new templates. By default this is the system's locale. The locale can be changed per template; see "Locale" on page 336.
  - Select System Locale to use the operating system's locale settings.
  - Select Explicit Locale to choose a static locale from the drop-down list.

Log Settings

PrintShop Mail Connect logs the activities it undertakes whilst running. New Connect logs are created daily and are held for a period before they are automatically deleted.

The settings on this page determine the level of logging and how long the log files should be retained.

These log files can be an essential resource when diagnosing issues with OL Support. The logging settings are global to ***all*** Connect applications and the Logging Preferences can be adjusted from within any of the Designer and Print Manager dialogs.

The settings are:

- **Overall Logging Level** selection: Select the overall Connect Logging level. This controls how much logging Connect will do.
  By default the logging level is set at the midpoint of *Info*, but it can be set higher to include more logging (*All*, *Trace*, *Debug*), or lower to reduce the amount of logging (*Warning*, *Error*).
  The logging is hierarchical, with each Logging Level selection containing all of the Logging Levels below it. The default logging *Info* Level contains all *Error* and *Warning* entries plus informational log entries.

> **Warning**
>
> Higher logging settings will have an impact upon Connect production speeds, as well as leading to substantially larger log files.
>
> We recommended leaving the logging level to *Info* and only using the higher levels of

ogging in conjunction with advice from OL support.

- **Rollover policy** selection: Chose whether to retain Connect log files for a certain number of days (**Daily logs**) or based upon some predetermined hard disk usage limitations (**Size-based logs**). The selection are as follows:
  - **Daily Logs**: Use this setting to determine how many days Connect Logs are to be kept, before they are deleted.
    - **Number of days to retain logs**: This value only impacts upon historic (closed) logs.
      Chose between 1 and 99,999 days.
      The default value is set to 10 days for a new installation and 99,999 days on existing installations (to preserve backward compatibility).
  - **Size-based logs**: Use this setting to restrict log file size, and to keep only a specified number of them.
    By combining the maximize individual log file size with the amount of log files to retain, this effectively allows a hard disk space usage limitation to be placed upon the logging process.
    - **Maximum size for log file**: This sets the maximum size a log file can reach before the logging system creates a new file.
    - **Number of files to keep**:This sets the maximum number of log files kept in the log folder. The default value is set to 50 for a new Connect installation and 99, 999 for an existing installation (to preserve backward compatibility).
- **Logging pattern** edit box: This edit box determines the formatting of the individual log entries.
  By default it is set to a date/time value which allows for simple searching within log files.

> **Warning**
>
> We recommend leaving the Logging pattern to the default value.
>
> If you do need to change the Logging pattern, please see the Pattern Formatting

guide (https://logback.qos.ch/manual/layouts.html#conversionWord) for help in doing so.

- **Log message preview** display: This displays a real time example of the format and content of individual log file entries, based upon the **Logging pattern** setting.

Advanced Log Settings

The Advanced settings over-ride the Overall logging settings, and provide a greater level of logging granularity.

> **Warning**
>
> Higher logging settings will have an impact upon Connect production speeds, as well as leading to substantially larger log files.
>
> The Advanced Log Settings should only be set in conjunction with advice from OL support, to ensure that only the most relevant settings are set to the higher logging levels.

This Preferences page allows you to add ( ) or remove ( ) individual Connect Packages, or change their logging settings ( ).

SNMP Connection Settings

Options for printers that support Simple Network Management Protocol (SNMP) connection. Not all IPDS printers do, but most PostScript and PCL printers will. This protocol allows the printers to communicate their status to the Print Manager.

All the SNMP settings relate to SNMP connection and re-connection attempts. They are:

- **Connection attempts** entry: Defines the number of times the Print Manager will attempt to connect to a printer before abandoning the attempt. This setting applies only when the Print Manager is started.
- **Polling interval (secs)** entry: Defines the polling interval in seconds.
  This determines how often the Print Manager attempts to communicate with the printer,

and applies to all communications with the printer. It applies both to the initial **Connect attempts** (setting the interval between those connection attempts) as well as how often the printer is polled after a connection has been established.
Each time the printer is polled the printer status is updated in the "Printer Status pane" on page 1052.

- **SNMP keep retrying after timeout** checkbox. Tick this if you would like to attempt to reconnect to the printer if a connection fails (or times out). Once ticked, you can set the retry interval amount.
    - **Retry interval on Timeout (secs)** entry: Enter the length of time (in seconds) between connection retries. This interval period applies only to attempts at reconnecting to printers.

PrintShop Mail Connect Release Notes

This chapter contains the current PrintShop Mail Connect 2022.1.x Release Notes, as well as the Release Notes from previous versions.

- "OL PrintShop Mail Connect Release Notes 2022.1.5" below

Previous releases:

- "OL PrintShop Mail Connect Release Notes 2021.2.1" on page 1079
- "OL PrintShop Mail Connect Release Notes 2021.1" on page 1089
- "OL PrintShop Mail ConnectRelease Notes 2020.2.1" on page 1096
- "OL PrintShop Mail Connect Release Notes 2020.1" on page 1109
- "OL PrintShop Mail Connect Release Notes 2019.2" on page 1117
- "OL PrintShop Mail Connect Release Notes 2019.1" on page 1135
- "PrintShop Mail Connect Release Notes 2018.2.1" on page 1151
- "PrintShop Mail Connect Release Notes 2018.1.6" on page 1178
- "PrintShop Mail Connect Release Notes 1.8" on page 1201
- "PrintShop Mail Connect Release Notes 1.7.1" on page 1222
- "PrintShop Mail Connect Release Notes 1.6.1" on page 1243
- "PrintShop Mail Connect Release Notes 1.5" on page 1251
- "PrintShop Mail Connect Release Notes 1.4.2" on page 1258

OL PrintShop Mail Connect Release Notes 2022.1.5

License Update Required for Upgrade to OL Connect 2022.x

From PrintShop Mail Connect 2019.1 onwards, only customers with a current OL Care subscription will be able to update Connect.

Every Connect license now contains an end date that represents the last day of OL Care coverage for each customer. Upgrades and updates of Connect will be freely available up until that end date, but will not be available thereafter, unless the OL Care period is extended.

This new licensing model requires anyone upgrading from Connect versions earlier than

2019.1 to first update their PrintShop Mail Connect License.

A dialog box appears as part of the Connect upgrade process requesting you to do this.

It is heavily recommended that you first update the OL Update Manager, as newer versions of the Update Manager can update your PrintShop Mail Connect License to the required version, then install Connect 2022.1.

For further details on how to upgrade the Update Client and update your Connect License see the Update Manager 1.5 - Upgrade Guide.

If issues arise, or if you need to renew your OL Care subscription, please contact your local Customer Care group.

Backup before Upgrading

It is recommended that you always backup your existing Connect preferences before upgrading to a new version. This will enable you to revert back to the previous version, in a worst case scenario in which the new version introduces issues with your existing production processes. Whilst the probability of such a worst case scenario is remote, it cannot hurt to take some simple precautions, just in case.

For instructions on how to do so, please see how to backup an existing Connect installation.

Overview

This document provides an overview of the new features and enhancements in PrintShop Mail Connect2022.1, as well as some important installation information.

Installing OL Connect 2022.1

- PrintShop Mail Connect is released as a 64 Bit version only.
- Full details on installing (Installation Wizard) and licensing (Activating a License) PrintShop Mail Connect can be found in the linked online Help pages.
- Note that PrintShop Mail Connect comes with a 30 day trial license by default.

Connect installer requires Administrator rights

Please note that the PrintShop Mail Connect installations can *only be run by users who have Administrator rights*.

OL Connect 2022.1.5 Improvements

Scaled barcode issue

It was discovered that some PrintShop Mail Connect 2022.1.x scaled barcodes could be hard to scan. This affected "small" scaled (fixed to width) PNG barcodes in particular. This issue has now been fixed. (87162)

OL Connect 2022.1.4 Improvements

Windows 11 and Windows Server 2022 now supported

Both PrintShop Mail Connect and PrintShopMail Workflow have now undergone thorough testing under the newest Microsoft Windows versions.

Having passed this testing process both PrintShop Mail 2022.1.4 and PrintShopMail Workflow 2022.1.4 are now officially supported under **Windows 11** and **Windows Server 2022** operating systems.

Additional improvements made to Connect 2022.1.x installer.

- The language chosen for the installer is now applied to the installed products. (86545)
- The 'Select Setup Language' dialog language and the default language selection now use the local machine language. (86623)
- Fixed some German language translations, including one which that caused issues with the installation. (86317/86582)
- Fixed problems with non-ASCII text corrupting the Connect Service installation. (86436)
- Cleaned up text displayed on 'Installation Finished' page. (86237)

OL Connect 2022.1.3 Improvements

Add support for text extraction on CID fonts that rely on external Adobe "-UCS2" CMap files. (86069)

OL Connect 2022.1.2 Improvements

Additional security feature added, making Connect 2022.1.2 the first official Connect 2022.1.x release. (86250)

OL Connect 2022.1.1 Improvements

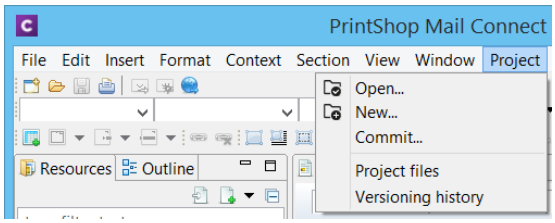
Some issues were discovered with the new Connect Installer during 2022.1.0 Release Candidate testing. These issues have now been fixed. (86124)

Version control added

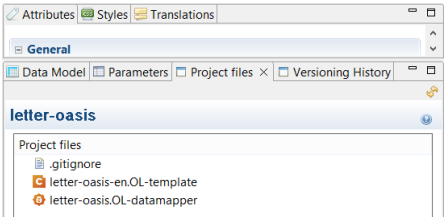
The **Designer** now uses Git integration to maintain a history of files inside named **Projects**. (81457)

This feature allows you to combine all the files used in a Project and keep a record of any/all changes made to any of the files within the Project thereafter.

Projects are added via new **Project** menu.

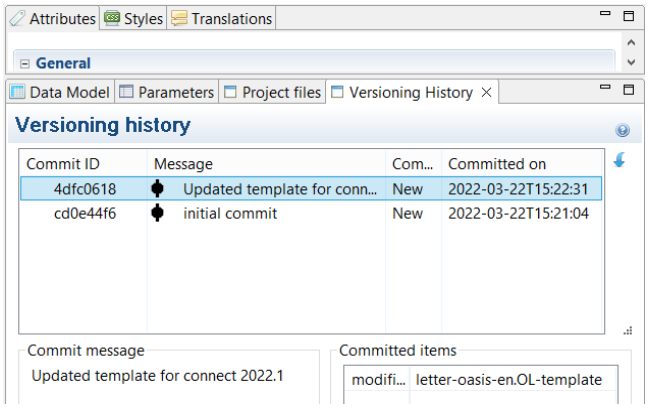


The files comprising a Project can be viewed in the new **Project Files** panel.



Each time you *commit* a resource (i.e., when you save an updated version of it), any additional information provided gets recorded alongside the file.

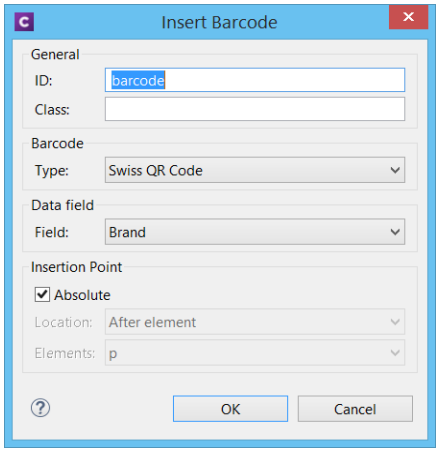
You can browse through the history of all changes in the **Versioning History** panel, which contains a complete record of who did what, on which date, and for which reason.



You can elect to revert to a version from within the history if something turns out to be flawed in your current project.

Swiss QR barcode

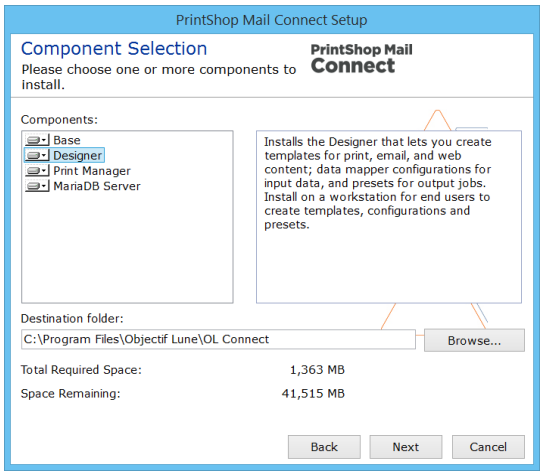
Swiss QR barcode support added to Connect. (84012)



New Connect Installer

The OL Connect installer has been totally re-engineered. (81627)

The new installer is based upon more standard architecture and supports rollbacks in case something goes wrong during installation.



Maria DB installed with Connect

Fresh installations of Connect now install **MariaDB**, rather than MySQL. Pre-existing Connect installations that had the Connect MySQL component installed will be migrated to MariaDB at

installation time.

Connect will continue to work with any customer provided MySQL databases that were present prior to Connect being installed. (83135)

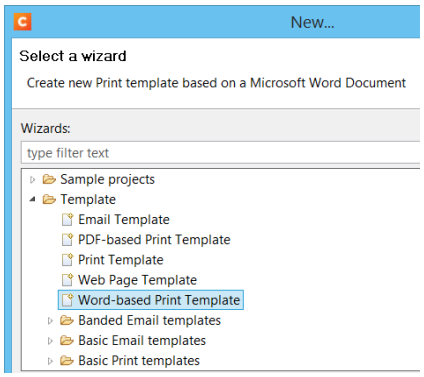
> **Note**
>
> The migration process might take some considerable time (many minutes), dependent upon the size of the existing Connect MySQL database.

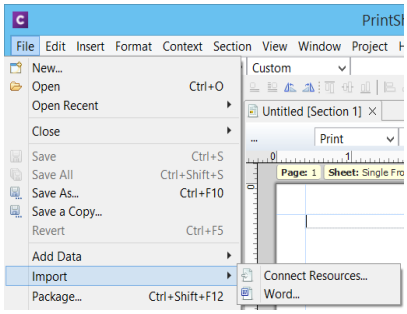
Java update

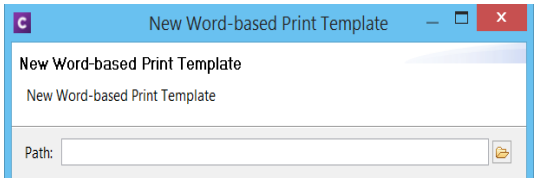
Updated the Java platform upon which Connect runs to AdoptOpenJDK 11.0.13+8 (82163)

Import Microsoft Word documents into print Templates

Added the ability to create new print Templates from Word documents and to import Word documents into an existing Templates. (83760)







The import uses the original Word document page size, margins, text formatting, lists, graphics, TextBoxes and even mail-merge information.

The contents of the Word file are added to the first Section in the Connect Template.
Mail merge place holders in Word documents are used to create a datamodel and userscritps.
Images are written to the Images folder and a base stylesheet is added to the Stylesheets panel. TextBox objects are converted to elements.

HTML templating using Handlebars

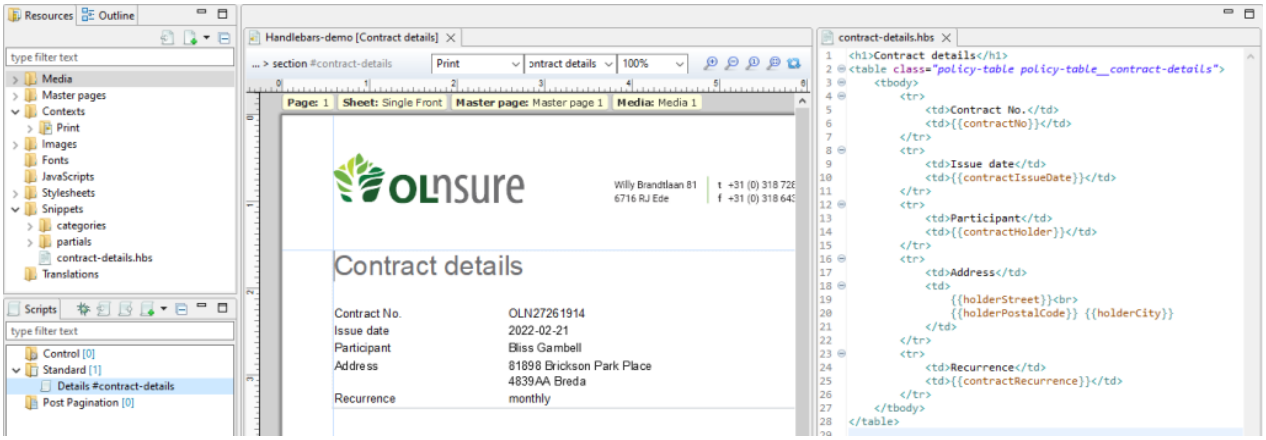
Provided new functionality that reduces the amount of scripting needed to create placeholder heavy and modular design templates. (83543)

If you've ever created OL Connect Designer templates for long documents like contracts and insurance policies, you're undoubtedly familiar with snippets. Snippet-heavy templates can become hard to maintain when each snippet requires multiple scripts to apply personalization and conditions. It requires the snippets and scripts to be thoughtfully organized (grouping,

naming conventions, search optimizations). But even then, things can become cluttered and may even impact the overall performance of your template.

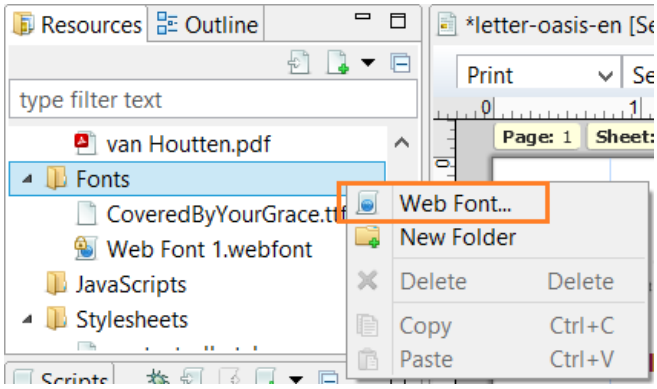
Users with a background in web design may be familiar with the Handlebars library. It is a popular templating library that provides a simple way to merge HTML snippets with data. This library has now been added to OL Connect Designer.

The Handlebars template snippets in OL Connect Designer will vastly reduce the number of scripts for the mentioned document types and help keep templates lean by creating semantic snippets.



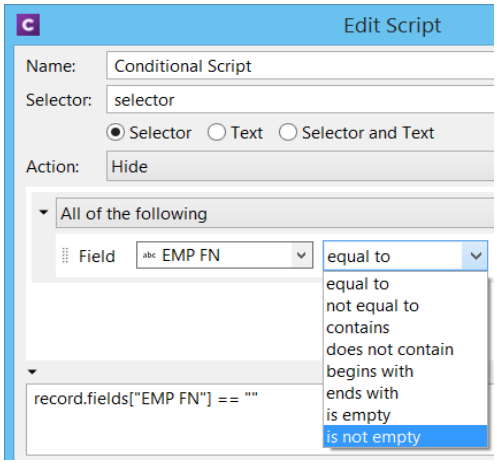
Improved Web Font support

The OL Connect Designer already supported web fonts, but only via CSS Stylesheets.
To simplify this, an option has been added to the Fonts folder to directly add remote fonts. (78790)



Conditional Script improvements

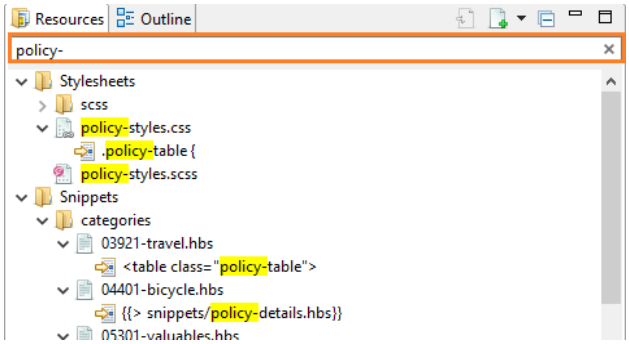
To further simplify Conditional Scripts, we have added two new conditions `is empty` and `is not empty`. (81585).



Search across Resources

The **Resources** panel has been expanded with a filter option. (83845)

This allows searching across master pages, sections, stylesheets and snippets.
The search results are grouped into the files containing the search term showing a preview and indicator of the term in each file.

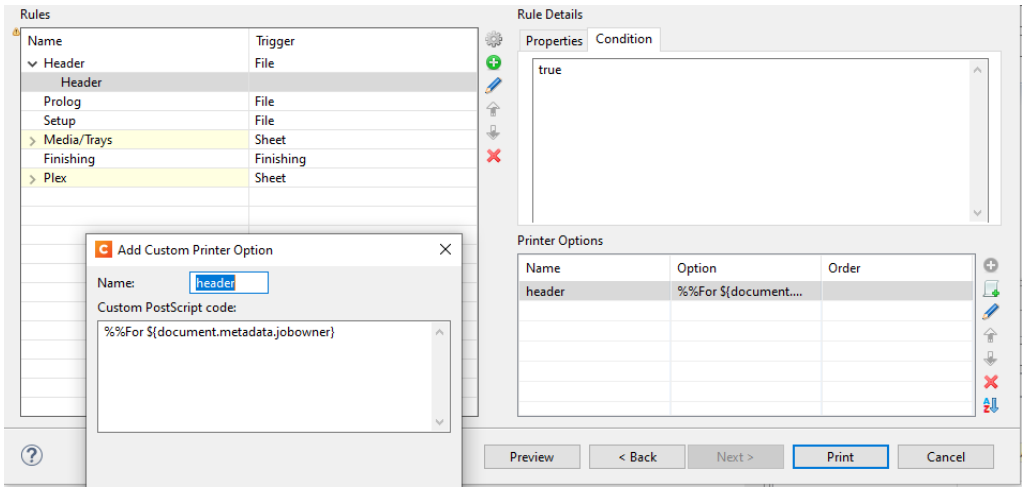


General Designer Improvements

- Fixed a problem with HTML snippets saved in source mode not actually saving the changes. (80971)
- Refresh types button in text scripts now enabled when needed. (82014)
- Templates are now still considered "changed" after making changes, saving and then undoing those changes. (83254)
- Connect will now continue to recognize remote translation files after a template is saved, closed and then reopened. (84825)
- Updated 3rd party library to fix issues with **USPS IMb** barcodes. (84847)

Tweak Postscript Headers

Raw Postscript can now be inserted at the header level. (83380)
Certain printers require very specific information in their header comments to work correctly. Although this could already be handled through custom Printer Definitions, it can now also be done directly through Dynamic PPD mode.



The inserts are added in the order they appear.

Use system fonts as fallback for missing input file fonts

When fonts are missing in input files (PDF, PostScript, etc.), it is now possible to have Connect use system fonts if the same fonts are installed locally. (54777)

At the moment this a technical preview feature, which requires manually editing the OL-outputpreset file, thus:

```
<fontSubstitutionConfig>
<pluginId>0</pluginId>
<useSystemFonts>true</useSystemFonts>
</fontSubstitutionConfig>
```

General Output Improvements

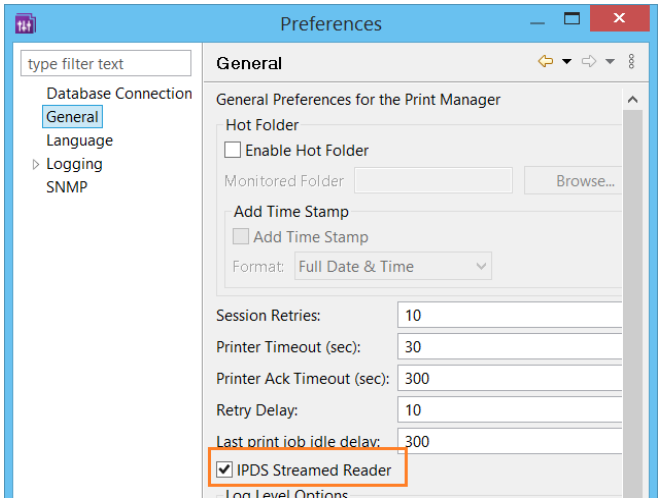
- Improved error/warning messaging. (75666)
- When the output engine flattening a transparency in a PDF file takes more than a minute, it will now logs messages to report progress on the flattening operation. (81683)
- It is now possible to change the default SMTP timeouts of the Merge Engine via setting system properties in the ini file. (82202)
- Added support for web font files (*.woff) in AddText components of custom Enhance workflows. (83567\83587)
- Prevented an "Attempt to paginate an already paginated DOM" error that could occur when scripts called merge.section.paginate() function. (84276)
- Improved handling of Open Type Format (OTF) fonts. (84502)
- Updated 3rd party library to fix issues with **USPS IMb** barcodes in output. (85117)

Print Output Improvements

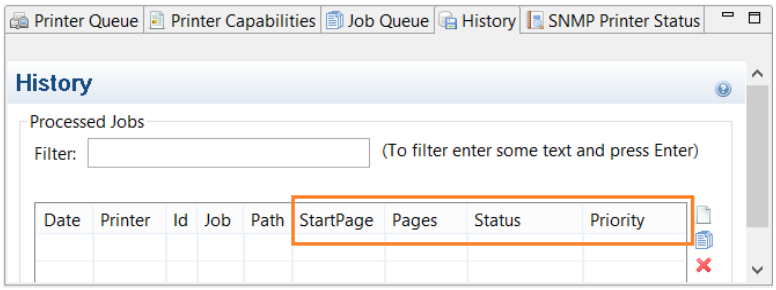
- Fixed issue in Dynamic PPD to include missing %EndPageSetup. (83388)
- Fixed an error occurring in PDF output for certain types of images when image optimization is enabled (Indexed with a base colorspace that is not DeviceRGB/Gray/CMYK). (83559)
- Media items in DynamicPPD Output presets are now optional. (83827)
- Crop marks were incorrectly positioned on the back side of duplex output sheets when using 90 or 270 degrees rotated impositioning. This has been fixed. (84770)
- Improved handling of PDFs with Transparency Groups when using Imposition. (84914)
- **AFD and IPDS** improvements
    - IPDS jobs that contained many barcodes were failing to print. The handling of barcodes has been changed to address this issue. (83927)
    - IPDS and AFPDS advanced logging now contains extra information on forms and images used. (84202)
    - Updated handling of PDF with Transparency Groups when using Imposition in IPDS output. This has resulted in improved processing times and smaller spool-file output. (84914)
    - To address an issue with IPDS Duplex printing starting out of sync after a restart, we now include Plex commands at the start of each physical sheet. (84916)

OL Connect 2022.1 Print Manager Improvements

- A problem was encountered on some IPDS printers, which would freeze after a period of time. This was due to issues with reading responses from the printer. A new "*IPDS streamed Reader*" preference has been added to the **Print Manager Preferences**, to cater for printers that experience this issue. (84485)



- Extra information now captured and displayed in the PrintManager. The extra information is available in the **History** and **Preview** panes. (85054)



OL Connect 2022.1 Improvements

Security and Installer improvements

- Connect dependencies have been updated. (83698)
- The Adobe PDF Library and PDF Converter libraries were updated to their latest version. (83712)
- The OL Update Manager included as part of the Connect has been updated to version 1.5 in Connect 2022.1. (85508)
- Updated Windows Service Wrapper library to improve security. (85076)
- NodeJS log lib (log4js) updated to improve security (85078)
- Upgraded dependencies to protect against Spring vulnerability CVE-2022-22965. (85809)

Known Issues

As of PrintShop Mail Connect 2018.1 the Known Issues have moved from the Release Notes to the Connect online Help.

They can be found here.

Previous Releases

OL PrintShop Mail Connect Release Notes 2021.2.1

License Update Required for Upgrade to OL Connect 2021.x

From OL Connect 2019.1 onwards, only customers with a current OL Care subscription will be able to update Connect.

Every Connect license now contains an end date that represents the last day of OL Care coverage for each customer. Upgrades and updates of Connect will be freely available up until that end date, but will not be available thereafter, unless the OL Care period is extended.

This new licensing model requires anyone upgrading from Connect versions earlier than 2019.1 to first update their OL Connect License.

A dialog box appears as part of the Connect upgrade process requesting you to do this.

It is heavily recommended that you first update the OL Update Manager, as newer versions of the Update Manager can update your OL License to the required version, then install Connect 2022.1.

For further details on how to upgrade the Update Client and update your Connect License see the Update Manager 1.3 - Upgrade Guide.

If issues arise, or if you need to renew your OL Care subscription, please contact your local Customer Care group.

Backup before Upgrading

It is recommended that you always backup your existing Connect preferences before upgrading to a new version. This will enable you to revert back to the previous version, in a worst case scenario in which the new version introduces issues with your existing production processes. Whilst the probability of such a worst case scenario is remote, it cannot hurt to take some simple precautions, just in case.

For instructions on how to do so, please see how to backup an existing Connect installation.

Overview

This document provides an overview of the new features and enhancements in PrintShop Mail Connect2022.1, as well as some important installation information.

Installing OL Connect 2022.1

- PrintShop Mail Connect is released as a 64 Bit version only.
- Full details on installing and licensing PrintShop Mail Connect can be found in the online help, which can be accessed from the software and the installer.
- Note that PrintShop Mail Connect comes with a 30 day trial license by default.

Connect installation requires Administrator rights

Please note that the PrintShop Mail Connect installations can *only be run by users who have Administrator rights*.

OL Connect 2021.2.1 Improvements

Fixed Fatal IPDS errors

IPDS jobs that contained many barcodes were failing to print. The handling of barcodes has been changed to address this issue. (SHARED-83927)

Fixed LPR printing errors

"*Timeout errors*" could sometimes be encountered when connecting to a remote printer from Connect server. The timeout values have now been optimized to fix this issue. (SHARED-83532)

OL Connect 2022.1 Designer Improvements

Sheet Configuration Repeats added

Sheet configuration can now be repeated after a fixed number of sheets.

For example when a script adds a number of paragraphs from a snippet based on the record index, the number of pages and sheets will increase when browsing through the records. (SHARED-82191)

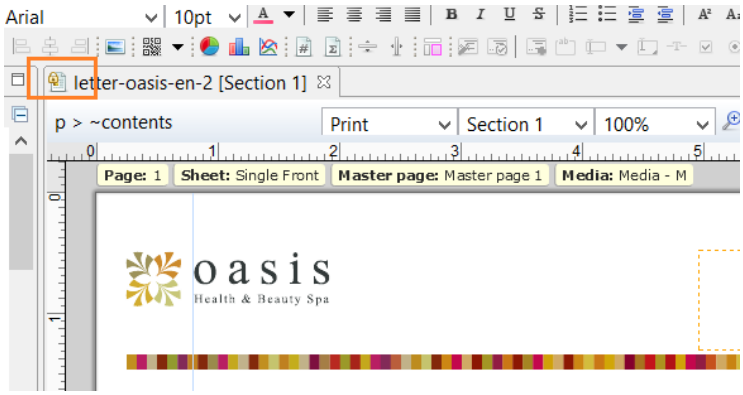
Retrieve a list of image resources

Added an option to retrieve a list of image resources via user/control scripting. This caters for scenarios such as validating whether a certain image exists in a template before setting the `src` attribute of an element in script. (SHARED-82303)

Improved handling of read-only Templates

Improved the handling on read-only OL-template files. (SHARED-82177)
Improvements include:

- Added a visual cue in Designer to indicate a Template is read-only.



- Saving changes in read-only files now launches the *Save as* dialog.

Retrieve the position information of elements

Introduced new script option `results.position()`, which can be used to retrieve the position (X and Y offset) of matched elements. This allows you to determine locations on the page and dynamically perform actions based upon the positions. (SHARED-81098)

Improved Drag and Drop support in data model fields

It's now possible to drag scripts with simple class and/or ID selectors to the template, similar to how this was already possible for scripts with text based selectors.

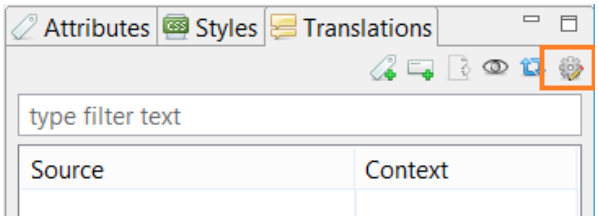
Dragging a data field from the data model to the scripts folder will create a script with a class based selector.

Holding down the ALT-key while dropping the data field will create a scripts with a text based selector (similar to the behaviour when dragging a data field into the template)

Also fixed a bug where no table was inserted when the dynamic table wizard was opened after ctrl-dragging a detail data field into the template. (SHARED-82152)

General Designer Improvements

- Improved translations throughout. (SHARED-81664)
- A toolbar button has been added to the **Translations View** to open the **Edit Locale Dialog** directly. (SHARED-81316)



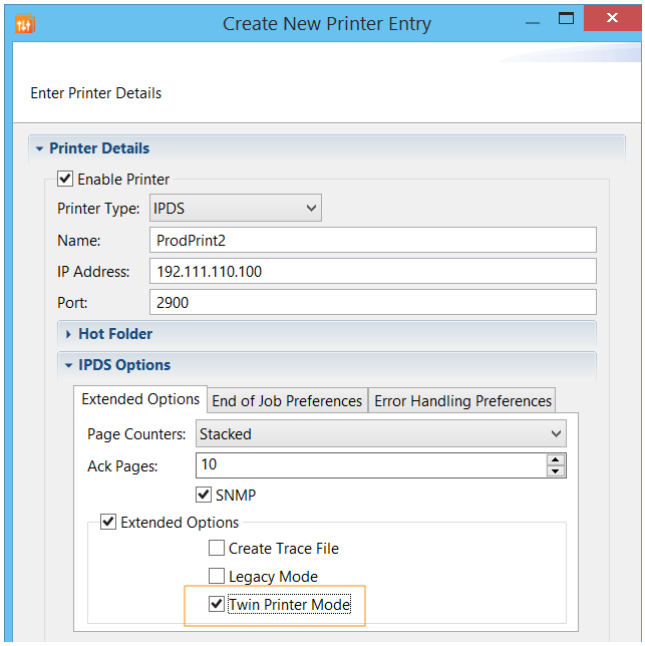
- When navigating through records in the Data Model view, the first click of an arrow was ignored. This has now been fixed. (SHARED-79830)
- You can now add a custom Template property by clicking the first empty line in the Custom properties dialog (File > Properties ... > Custom). (SHARED-80775)
- **Packaging** a Template no longer throws an error when a section that a thumbnail is created for is disabled by script. The packaged Template will now use the disabled section. (SHARED-81237)
- Fixed an issue with the formatting of default values of date fields, which caused errors when the Template was re-opened. (SHARED-81721)
- Fixed issues with URIs containing HTML encoded backslashes in the file path. (SHARED-81887)

- Prevented an error after saving a resource while a section with shared content is active. (SHARED-82200)
- The execution scope of a script folder now properly inherits the execution scope of its parent folder. (SHARED-82026)
- Fixed errors that could be encountered when previewing a template from Designer (using the Preview HTML button). (SHARED-82532)
- The Section properties dialog and the scripting API now both allow the width and height of a background image to be scaled independently. (SHARED-82616)
- When a data field is dragged to the email fields in an email section, the dragged field will now be used in the script generated. (SHARED-82876)
- Improved the way translation files are matched to the active locale. (SHARED-83079)
- The default insertion method for new text scripts has reverted to HTML, rather than text. (SHARED-83412)

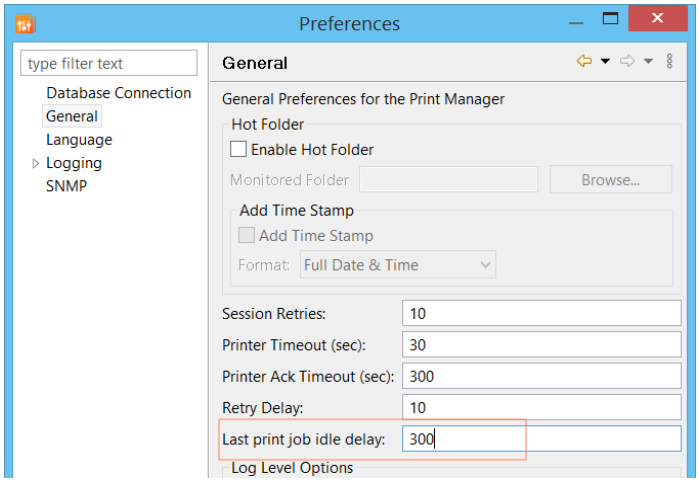
OL Connect 2022.1 Output Improvements

IPDS duplex 2-Up printing improvements

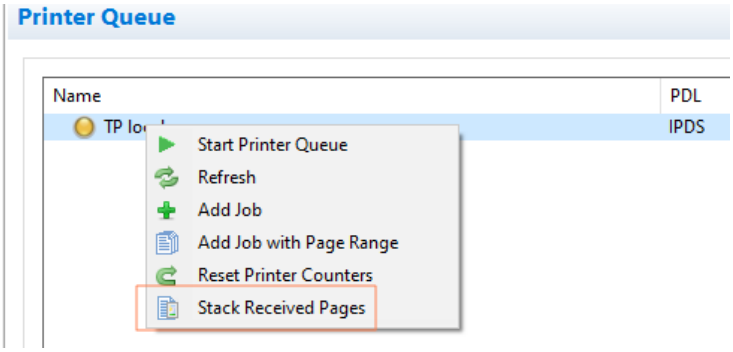
- Fixed a page size issue in Content generated by DataMapper. (SHARED-81960)
- Added support for selecting Twin System IPDS printers in **Print Manager**. (SHARED-82088)



- Added a **Print Manager** option to specify how long to keep a session open, once the last print job finishes.



- A new **Print Manager** option has been added to allow the printer to push all committed pages to the stack.



Improved password protected PDF output

- Fixed an issue with PDF output unnecessarily prompting for a password when viewed in some Android and iOS browsers and apps. (SHARED-82931)
- Fixed an issue that could lead to blank pages on some Acrobat Readers (iOS, Android and Chrome Extension). (SHARED-83034)

General Output Improvements

- Fixed and issue with PCL input conversion. (SHARED-81616)
- Include the *file currently open for flattening* in debug log messages. (SHARED-81685)
- Fixed an issue where detail table data from a previous run/record showing up in the next run/record, when a scripted table was included that did not use the standard table expander. (SHARED-82466)
- Pagination errors could occur when large scripted tables did not use the standard table expander and were split across pages. These errors have been fixed. (SHARED-82198)
- Issues with the ascender height calculation of **IMB barcode** marks have been fixed and IMB barcodes now meet USPS specifications. (SHARED-80812/83082)
- The output engine now remembers files it has previously run through the transparency flattener, reusing the flattened file if possible, improving output speed. (SHARED-80935)

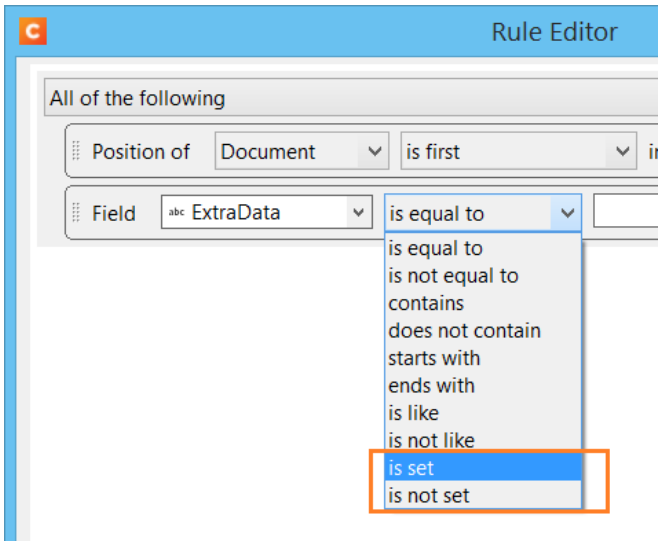
Email Output Improvements

- Fixed an issue whereby all web sections were attached to email instead of only the default web section. (SHARED-82986)

- Fixed an issue where elements with inline style `display:none` would be included in email content when the CSS inlining was not set to "*Apply CSS properties on elements*". (SHARED-82788)

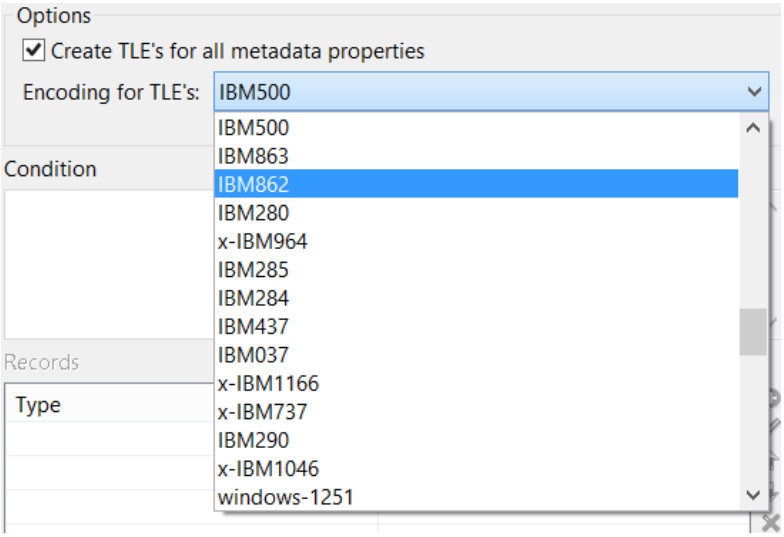
Print Output Improvements

- Added support for null comparisons in **Finishing Rules**.
  Two new data selection filter operators allow for detecting null data/param values in *data rules* and null/missing properties in *property rules*. The new operators are "*is set*" and "*is not set*". (SHARED-72938)



- The "*Job Size/Page Count*" and "*Speed/Overall Speed*" fields in the print summary in the "Connect Designer" logs and in the "Print" dialog are now being correctly reported. (SHARED-81385)
- A problem was fixed whereby image data from PNG files merged into a PDF were left uncompressed, increasing the file size. (SHARED-81804)
- Fixed issues with output presets edited in languages other than English. (SHARED-81843)
- Dynamic PPD printerdef and output presets were set to %%Title: ${template}. The %Title field now uses the jobName when using LPR. (SHARED-81958)
- Fixed pages size issue with DataMapper auto-generated Content Creation. (SHARED-81960)
- Improved Output Creation error messaging if the *impositioned pages won't fit on the sheet*. (SHARED-81962)

- Allow up to a 1pt deficit in media size when checking the required size for **impositioning**. This reduces errors if some lengths were specified in different units. (SHARED-82413)
- If the Binding type in a Dynamic PPD was set to Default, this would result in NULL errors when running jobs. This has been fixed. (SHARED-83310)
- **AFP and IPDS** improvements
  - When creating AFP/IPDS output with TrueType fonts, Connect uses Unicode values to refer to the glyphs. This would only work if there was a one-to-one mapping from Unicode to all the glyphs being used.
    Connect now fallbacks to using a bitmap font when unknown Glyphs or duplicate codes are encountered. (SHARED-80954)
  - Improved vector conversion in AFP and IPDS. (SHARED-82246)
  - Report errors and stop the job, when an AFP output rip fails. (SHARED-82814)
  - **TLE metadata** in AFP output now allows selection of character encoding type. (SHARED-82356)



  - When running an Enhance configuration in Connect, TLEs read from an AFP input file were duplicated in an AFP output file. This issue has been fixed. (SHARED-82619)

OL Connect 2022.1 Improvements

Security improvements

- Multi-line log messages converted to a single line, to preclude *CRLF Injection* exploits. (SHARED-74718)

Print Manager Improvements

- Improved translations throughout. (SHARED-81664)
- **Twin IPDS printer configurations** are now better supported. (SHARED-81955)
- Added warnings to the *Messages* pane when a spool file is being held by another process and cannot be processed. (SHARED-82885)

General OL Connect improvements

- Improved backend database startup process across Connect applications, reducing issues with startup failures in containerized environments. (SHARED-80360)

Known Issues

As of PrintShop Mail Connect 2018.1 the Known Issues have moved from the Release Notes to the Connect online Help.

They can be found here.

OL PrintShop Mail Connect Release Notes 2021.1

License Update Required for Upgrade to OL Connect 2021.x

From OL Connect 2019.1 onwards, only customers with a current OL Care subscription will be able to update Connect.

Every Connect license now contains an end date that represents the last day of OL Care coverage for each customer. Upgrades and updates of Connect will be freely available up until that end date, but will not be available thereafter, unless the OL Care period is extended.

This new licensing model requires anyone upgrading from Connect versions earlier than

2019.1 to first update their OL Connect License.

A dialog box appears as part of the Connect upgrade process requesting you to do this.

It is heavily recommended that you first update the OL Update Manager, as newer versions of the Update Manager can update your OL License to the required version, then install Connect 2021.1.

For further details on how to upgrade the Update Client and update your Connect License see the Update Manager 1.3 - Upgrade Guide.

If issues arise, or if you need to renew your OL Care subscription, please contact your local Customer Care group.

Backup before Upgrading

It is recommended that you always backup your existing Connect preferences before upgrading to a new version. This will enable you to revert back to the previous version, in a worst case scenario in which the new version introduces issues with your existing production processes. Whilst the probability of such a worst case scenario is remote, it cannot hurt to take some simple precautions, just in case.

For instructions on how to do so, please see how to backup an existing Connect installation.

Overview

This document provides an overview of the new features and enhancements in PrintShop Mail Connect 2021.1, as well as some important installation information.

Installing OL Connect 2021.1

- PrintShop Mail Connect is released as a 64 Bit version only.
- Full details on installing and licensing PrintShop Mail Connect can be found in the online help, which can be accessed from the software and the installer.
- Note that PrintShop Mail Connect comes with a 30 day trial license by default.

Connect installation requires Administrator rights

Please note that the PrintShop Mail Connect installations can *only be run by users who have Administrator rights*.

OL Connect 2021.1 Enhancements

OL Connect 2021.1.1

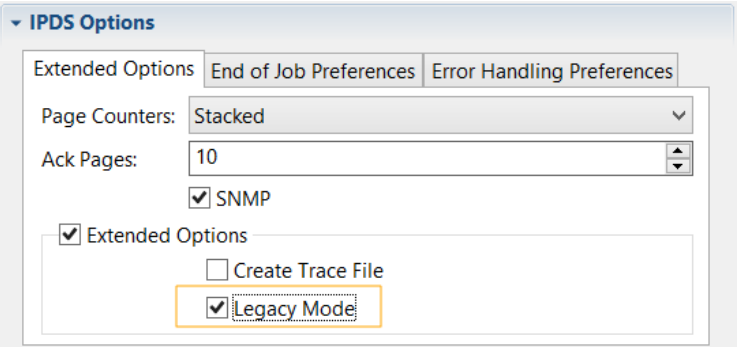
In the OL Connect 2021.1.0 partner preview it was discovered that dragging group data from the DataMapper into a template was sometimes throwing an error. This has been fixed in the 2021.1.1 patch release of Connect. (SHARED-82134)

Security Improvements

For increased security the cookie, localstorage and indexedDB features have now been disabled for the internal browser used by the Designer and Merge Engines. (SHARED-77792)

Print Manager Improvements

For IPDS printers, **Multi Job mode** is now set as the default. To disable Multi Job mode, select *Legacy mode* option. (SHARED-78768)



Installer Improvements

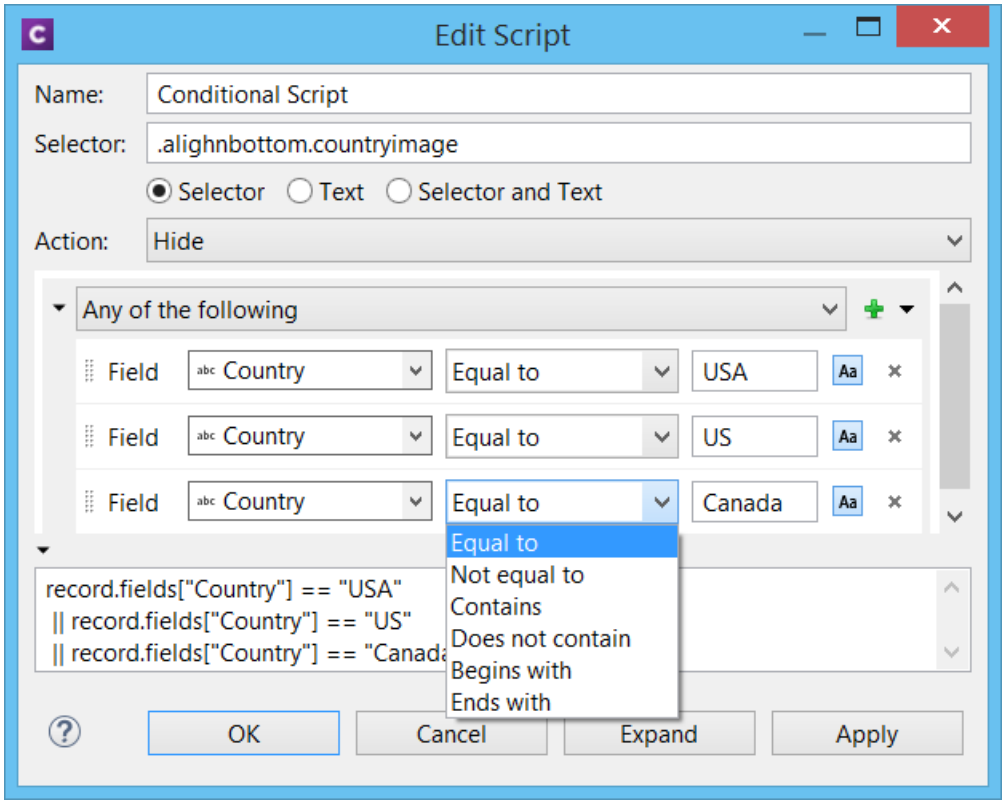
- Connect can now be installed when no internet access is available. (SHARED-79964)
- The issue with slow responsiveness in Installer component selection has been fixed. (SHARED-76902)
- On a Connect update/upgrade, the installer Welcome page now clearly displays both the currently installed version and the upgrade (update) version about to be installed. (SHARED-77819)
- You can now choose to omit the **Update Manager** from the installation, using the command line parameter `UPDATEMANAGER=0`. (SHARED-79761)
- Fixed an uninstallation issue caused by some Connect temporary files being set as read-only. (SHARED-79765/79838)

- Fixed an installer issue that was preventing Connect from installing on Hyper-V (and some other virtual environments other than VMware). (SHARED-79918)

OL Connect 2021.1 Designer Improvements

Conditional Wizards

The Designer conditional Wizard for elements and the Wizard for conditional print sections have been dramatically improved, allowing very complex rules to be created without any need for scripting. (SHARED-79806)



Scripting Improvements

- The default insert method for new search-and-replace scripts is now "Text" instead of "Html", to improve performance. (SHARED-80376)
- Fixed syntax highlighting issue with the Script keyword '*let*' . (SHARED-78644)
- We now generate an error if a script tries to move an element relative to itself (which Connect does not support). (SHARED-79945)
- Fixed an issue where an incorrect selector would be generated for scripts when a datafield containing spaces in the name is dragged from the data model to the template. (SHARED-80535)
- Fixed a problem with UNC paths in the Dynamic Image Script wizard. (SHARED-81145)

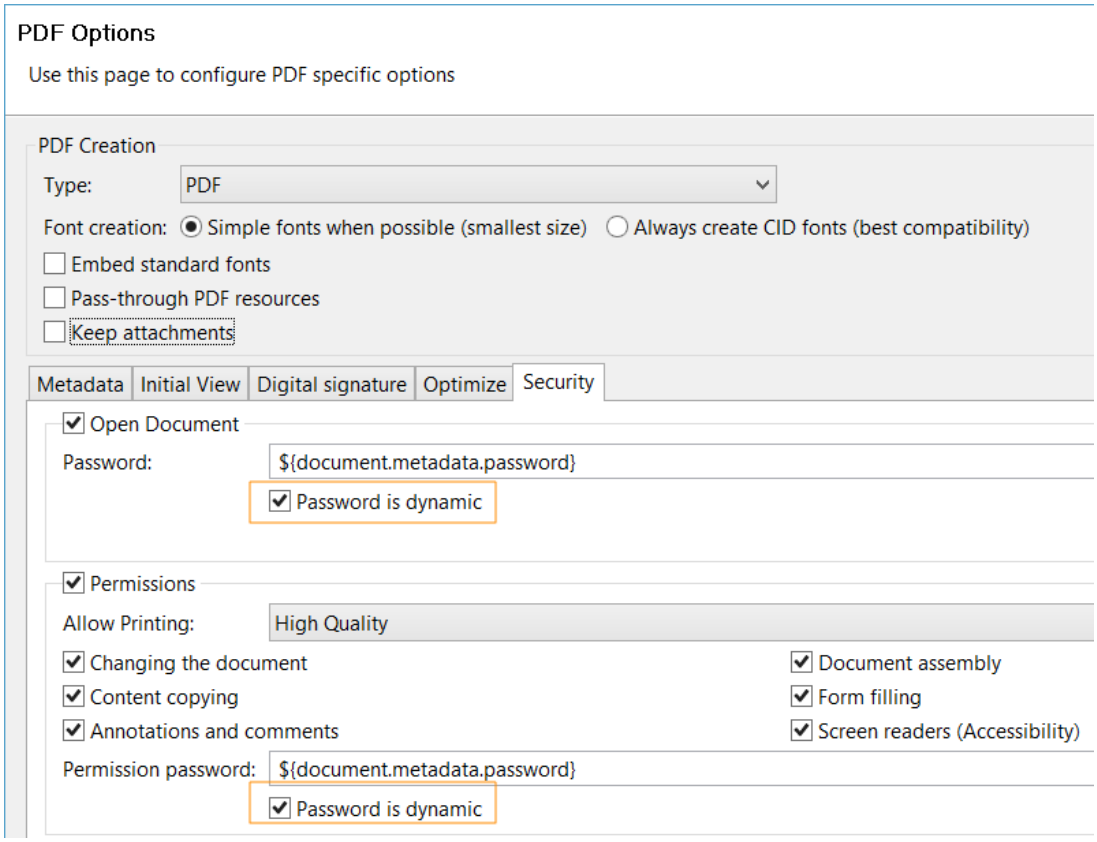
General Designer Improvements

- Improved Designer translations/localization. (SHARED-76798)
- **JSON editors** and the `loadjson()` function now support UTF-16 and UTF-32, in addition to UTF-8. (SHARED-74159)
- After adding an **HTML snippet** you are now prompted to select encoding (UTF-8 by default). (SHARED-78552)
- Added validation to **JSON resource** editors. (SHARED-79937)
- New prompt added when dragging a field from a detail table to a location outside a detail table. (SHARED-80116)
- Add support for copying contents of an extraction field into the clipboard. (SHARED-80164)
- Can now set the Virtual Stationery of a media back to "None" by using the clear button. (SHARED-78802)
- Fixed an issue within the Source tab, whereby HTML in the source tab unintentionally contained helper elements for resizing and dragging. (SHARED-78928)
- Fixed an issue with converting a paragraph to a header1 (p to h1), in which the parameter ID would be lost. (SHARED-79022)
- Fixed an issue where the cellpadding and cellspacing values that were set directly on a table were removed after switching to Preview, if these values were not defined in stylesheets. (SHARED-79543)
- Fixed a problem with scripts referencing `BackgroundResource.RESOURCE_PDF` producing an error. (SHARED-79606)
- Fixed issue with the popup menu in the breadcrumbs bar. (SHARED-79727)
- Show Edges settings once again persist through view changes. (SHARED-79970)
- Fixed a problem with absolute positioned elements nested inside other absolute positioned elements. (SHARED-80003)

OL Connect 2021.1 Output Improvements

Dynamic passwords for protected PDFs

The password for opening/protecting output PDF files can now be set dynamically. (SHARED-80537)



General Output Improvements

- The stability of the Output engine has been improved. Particularly when the engine is running with low memory. (SHARED-77203)
- The output engine is now more accepting of certain invalid CFF fonts (those containing hintmask without hstemvm/vstemvm). (SHARED-76464)
- Fixed an error in the output of Unicode (3,1) cmap tables for TrueType fonts which could trigger an exception if multiple glyphs have the same Unicode mapping. (SHARED-78284)
- Fixed issues with Type 1 and CFF font glyphs. (SHARED-79973)
- Versioning has been added to font mapping files for TextRestore. To prevent problems in text extraction going unnoticed, font mapping files without a version or with a mismatching

version will result in an error. Resolving those errors requires manual updating of these font mapping files. (SHARED-80568)

This change only applies to Enhance configurations run from Connect.

Print Output Improvements

- Fixed an issue with misleading warning messages for certain output presets when opening the production Print Wizard or Output Preset editor. (SHARED-74399)
- **Dynamic PPD rule validation** is now immediately updated after the additional elements for a selected PPD element are updated. (SHARED-77060)
- Restored **maximum stack depth** setting in the Imposition options page back to a maximum of 100,000. (SHARED-79357)
- Increased **maximum paper sizes** available for PostScript output. (SHARED-79433)
- The output engine now can process intermediate files (from merge engine) that are over 2 GBs in size. (SHARED-80490)
- Fixed an issue with **IPDS pages** larger than 22.75 inches failing to load on the printer. (SHARED-80846)
- For some output formats, such as PostScript, output creation didn't terminate for certain input files containing transparency, because Connect's transparency flattening tool was unable to process it. This has been fixed. (SHARED-81542)

Known Issues

As of PrintShop Mail Connect 2018.1 the Known Issues have moved from the Release Notes to the Connect online Help.

They can be found here.

OL PrintShop Mail ConnectRelease Notes 2020.2.1

License Update Required for Upgrade to Connect 2020.x

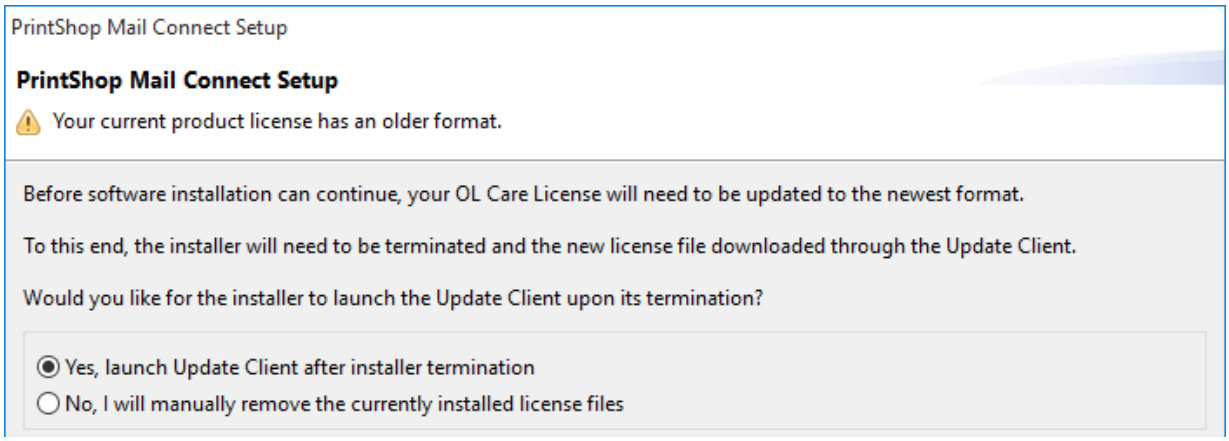
From OL Connect 2019.1 onwards, only customers with a current OL Care subscription will be able to update Connect.

Every Connect license now contains an end date that represents the last day of OL Care

coverage for each customer. Upgrades and updates of Connect will be freely available up until that end date, but will not be available thereafter, unless the OL Care period is extended.

This new licensing model requires anyone upgrading to 2020.2 from Connect versions earlier than 2019.1 to first update their OL Connect License.

A dialog box appears as part of the Connect upgrade process requesting you to do this.



It is heavily recommended that you first update the OL Update Manager, as newer versions of the Update Manager can update your OL License to the required version, then install Connect 2020.2.

For further details on how to upgrade the Update Client and update your Connect License see the Update Manager 1.3 - Upgrade Guide.

If issues arise, or if you need to renew your OL Care subscription, please contact your local Customer Care group.

Backup before Upgrading

It is recommended that you always backup your existing Connect preferences before upgrading to a new version. This will enable you to revert back to the previous version, in a worst case scenario in which the new version introduces issues with your existing production processes. Whilst the probability of such a worst case scenario is remote, it cannot hurt to take some simple precautions, just in case.

For instructions on how to do so, please see how to backup an existing Connect installation.

Overview

This document provides an overview of the new features and enhancements in PrintShop Mail Connect 2020.2, as well as some important installation information.

Installing PrintShop Mail Connect 2020.2

- PrintShop Mail Connect is released as a 64 Bit version only.
- Full details on installing and licensing PrintShop Mail Connect can be found in the online help, which can be accessed from the software and the installer.
- Note that PrintShop Mail Connect comes with a 30 day trial license by default.

Connect installation requires Administrator rights

Please note that the PrintShop Mail Connect installation can *only be run by users who have Administrator rights*.

OL Connect 2020.2.1 Enhancements

Fixed Connect 2020.2 Installation error

An installer issue that prevented Connect 2020.2 from installing on Hyper-V (and some other virtual environments other than VMware) has been fixed. (SHARED-79958)

Fixed Connect 2020.2 uninstallation error

It was discovered that the Connect 2020.2 Designer would not fully clean up temporary folders when the Table Wizard was used. This would lead to "*Failed to uninstall*" error messages during uninstallation, even though Connect would uninstall successfully. This issue has been fixed in Connect 2020.2.1. (SHARED-79922/79765)

Fixed issue with script references to BackgroundResource.RESOURCE_PDF

An issue was introduced in Connect 2020.2 whereby script references to BackgroundResource.RESOURCE_PDF would produce errors. This has been fixed in 2020.2.1. (SHARED-79927)

OL Connect 2020.2 Enhancements

OL Connect Security Improvements

- Security of the Weaver engine has been improved. (SHARED-74605)

> **Note**
>
> Important note for those running an Enhance config in which a JavaScript uses Java classes. JavaScripts evaluated by the Weaver engine no longer have access to the Java API by default. In order for such an Enhance config to run, any necessary Java classes must be whitelisted. For further details, please consult the Knowledge Base.

- Multiple PrintShop Mail Connect dependencies updated. (SHARED-76431)
- Improved support for encrypted email output (TLS/SSL). (SHARED-76460)

Print Manager Improvements

- Deleted printers were not cleared from the system and they could not be reintroduced. This has been fixed. (SHARED-41939)
- Standalone Print Manager installation now installs the Software Activation Tool. (SHARED-70967)
- Applying a job history limit will now take immediate effect. (SHARED-77071)
- Multi-job mode is the recommended setting for IPDS and it has now been made the default. (SHARED-77176)

General Improvements

- Cleaned up the error messages returned by Connect Server. Messages will no longer be duplicated, and they will not have exception class prefixes, such as "ApplicationException:". (SHARED-70963)
- Improved Server side error handling when a template fails to load. As part of this improvement we also now log the template name when a template is loaded. (SHARED-71024)

Installer Improvements

- Add a feature to the **silent installer** of Connect to allow specifying a target folder other than the default folder, using "*install.directory=xxx*". (SHARED-75669)
- **Silent installations** can now specify connecting to a database with an encrypted connection, using "*database.usessl*". (SHARED-75671)
- **Silent installations** no longer need a connection to a database to complete. (SHARED-75775)
- In the Connect installer, the **Messenger Service** is now only installed when the Designer module is selected. (SHARED-76189/76318)
- Improvements made for installations on machines where Connect failed to uninstall correctly. (SHARED-77148)

Connect 2020.2 Designer Improvements

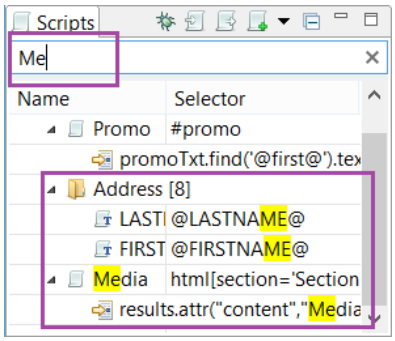
Improved Background Image options

- The **Section Background** functionality has been updated to support image types other than PDF. This includes .png, .jpg and .tiff (including multi-page TIFFs) files. This is also supported via the Control Script API.(SHARED-72730)
- In versions of Connect prior to 2020.2, if a page had no content except for a linked DataMapper background, then Connect would consider it an "empty" page when determining whether or not to "*Omit Master Page Back in case of an empty back page*" (available as an option in the sheet configuration of a section). This has now been fixed in Connect 2020.2, and such pages are no longer considered empty. (SHARED-72588)

> **Note**
>
> This can affect the output of existing templates.

Scripting Improvements

- Added ability to quickly search through script names and contents for specified text. (SHARED-60305)



- Auto suggestion improvements made for table properties in the JavaScript editor. (SHARED-70690)
- Introduced **prev()** and **next()** functions to the "*results*" object in our scripting API. (SHARED-70764)
- Fixed a problem with **Find Previous** in the script debugger. (SHARED-76737)
- Fixed minor issues in the script folder properties dialog. (SHARED-76996)

- Added the ability to retrieve the size of a print section or medium in any script, and the ability to change the size in a control script. (SHARED-77446)
- Fixed *results.add()* functionality. (SHARED-78447)

General Designer Improvements

- When the size of a print section is changed, you can now choose to "*Update linked media*" if the media is only used by that section. If it is not, then the option will not be available. (SHARED-65278)
- Holding shift while dragging boxes now drags the box in either horizontal or vertical directions. (SHARED-72158)
- Fixed failures with the '*Get Job Data File on Submit*' feature. (SHARED-72697)
- Fixed an issue with the *Export Report* function skipping Web information for some templates. (SHARED-73640)
- *Apply* button added to **Section Properties Dialogs**, which allows applying changes without closing the dialog. Clicking the *Cancel* button reverts any previously applied changes. (SHARED-74164)
- Fixed an issue with broken PDF hyperlinks that would occur when not all child elements of a hyperlink were on the same page. (SHARED-74482)
- Fixed some issues in the formatting dialog. (SHARED-74908)
- The case sensitive option of Conditional Scripts is now also available for HTML String fields. (SHARED-74970)
- Added support for multi-selections to the **Make Conditional** feature. (SHARED-76443)
- A change introduced in 2019.2 disallowed the selection of multiple elements in the Outline tree in Designer. This has now been remedied. (SHARED-76550)
- Fixed issue with *Save As...* option sometimes being disabled. (SHARED-76740)
- Fixed an issue whereby inserting a Table into a snippet would not mark that snippet as changed. (SHARED-76875)
- Introduced a new preference to control the behaviour when dragging data fields to the main editor. By default, scripts now use a class selector. (SHARED-76934)
- Fixed an issue where pagebreaks on the same page were ignored when a pagebreak was applied to the last row of a table. (SHARED-77946)
- Fixed an issue whereby some dates in JSON data would not resolve correctly in the Designer. (SHARED-78218)

Connect 2020.2 Output Improvements

2D Vector Barcode optimization

2D Scalable Vector Graphics (SVG) barcodes now optimized. (SHARED-77579)

This optimization includes these barcode types:

- Aztec Code
- DataMatrix
- QR Code
- Variations of the above, such as GS1 DataMatrix

True Type font handling

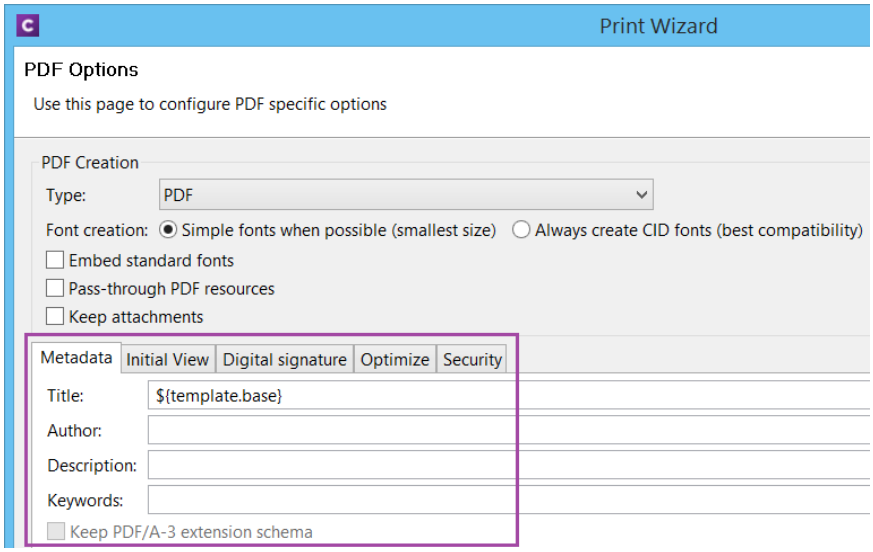
A major overhaul of TrueType font handling in the Connect output engine was conducted. This should help in reducing errors relating to both caching and parallel processing. (SHARED-61839)

PDF Digital Signature Improvements

- When creating a digitally signed PDF that includes a timestamp acquired from a timestamp server, Connect now automatically chooses the hash algorithm for the timestamp request.
  SHA-256 is tried first and if the timestamp server does not support it, Connect then tries SHA-384, SHA-512 or SHA-1 (in that order), defaulting to SHA-256 if none are supported. (SHARED-77850/78085)
- Fixed an issue whereby the use of certain certificates would result in incorrectly signed PDF output. (SHARED-77915)

PDF Output Improvements

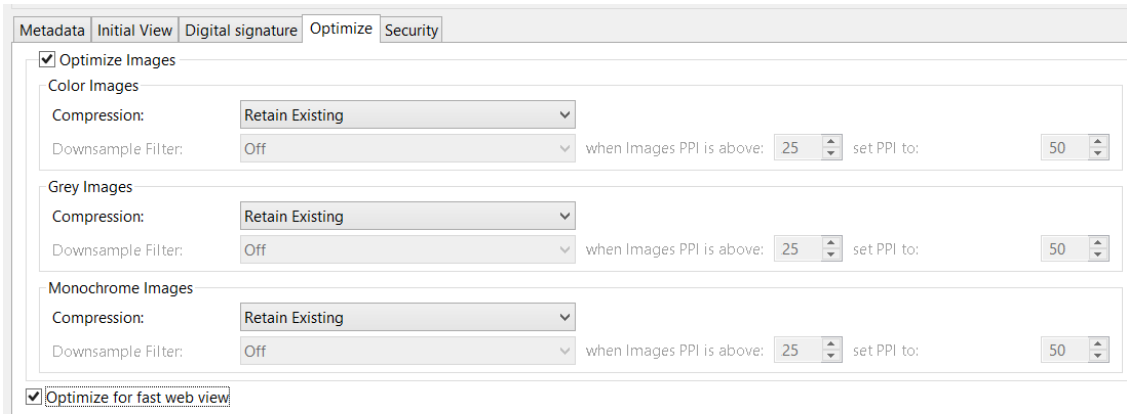
The PDF Output options have been substantially improved. This required a redesign of the **PDF Options** page, to provide all the new options on a single Wizard page. The various options have now been placed in logically distinct tabs.



The improvements include the following:

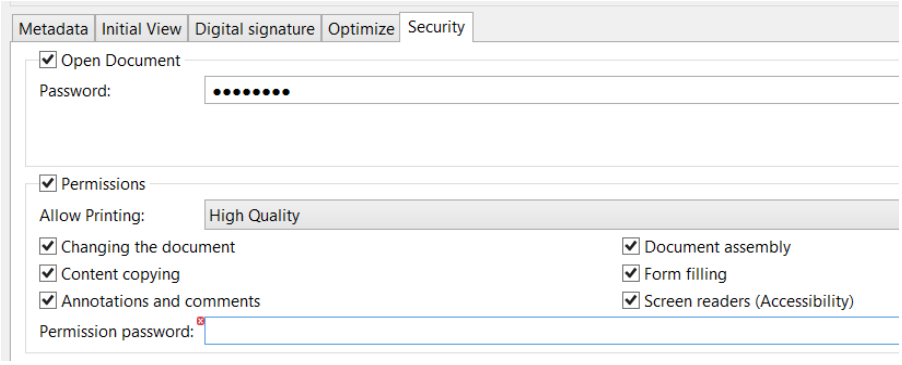
- New image optimization options added for down-sampling mono, greyscale, and color images. (SHARED-75219)

  This includes a new option to "*Optimize for Fast Web View*". (SHARED-75210)



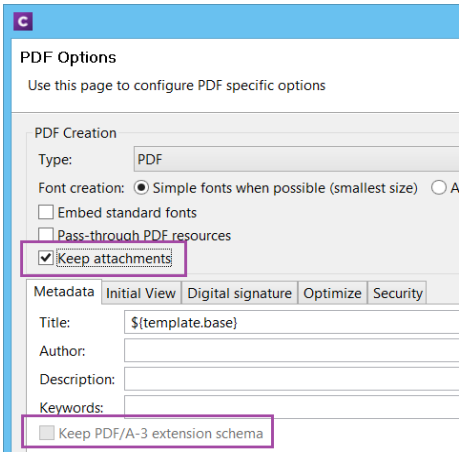
- Added PDF Security options. (SHARED-75213/75216)

  The Security options mirror those available in Adobe PDF applications.



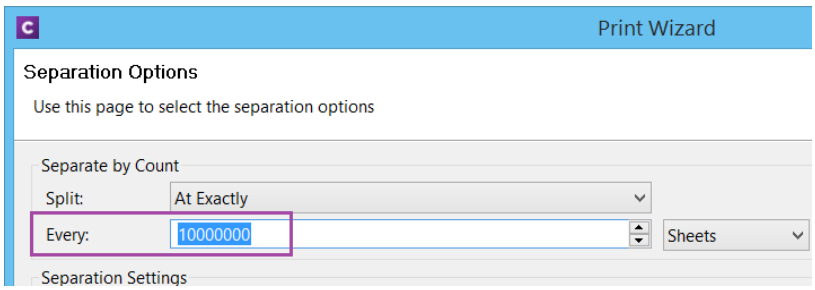
- Improved support for processing digitally signed PDF/A-3 attachments. (SHARED-77267/78255)

  You can now choose to *Keep attachments* and whether to retain PDF/A-3 extension schema metadata, in order to support production of digitally signed PDF/A-3 using that metadata.

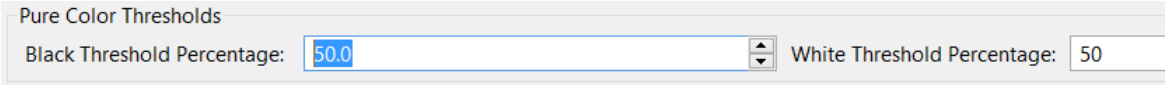


Print Output Improvements

- Fixed issues with missing glyphs and space characters in some double-byte jobs. (SHARED-74531)
- Placeholders for missing glyphs are now output as black, allowing output to remain monochrome. (SHARED-75513)
- Fixed issues with large double-byte font jobs. (SHARED-76730)
- The upper limit for the *Separation by Count* setting in the **Separation Options** Page has been increased to 10,000,000. (SHARED-77375)



- Optimized line drawing commands to reduce size of AFPDS output. (SHARED-77622)
- Update and improve **image support** in AFP, PCL and IPDS output. (SHARED-77716/77967/77972/77993)
- Added support for **Overlays** (these will be scaled to the needed size) to AFP output. (SHARED-77768)
- *Pattern support* setup code modified in PostScript output, so it will no longer interfere with custom *Page Device setup* from printer definitions.
- Fixed a license watermark issue in PostScript output with pages containing negative font size text. (SHARED-78076)
- Increased the limit of both **Black and White thresholds** to 50% for those print output types that support it. (SHARED-78298)



Email Output Improvements

- Improved error messaging for emails containing PDF attachments containing hyperlink images. (SHARED-76540)

Known Issues

As of PrintShop Mail Connect 2018.1 the Known Issues have moved from the Release Notes to the Connect online Help.

They can be found here.

OL PrintShop Mail Connect Release Notes 2020.1

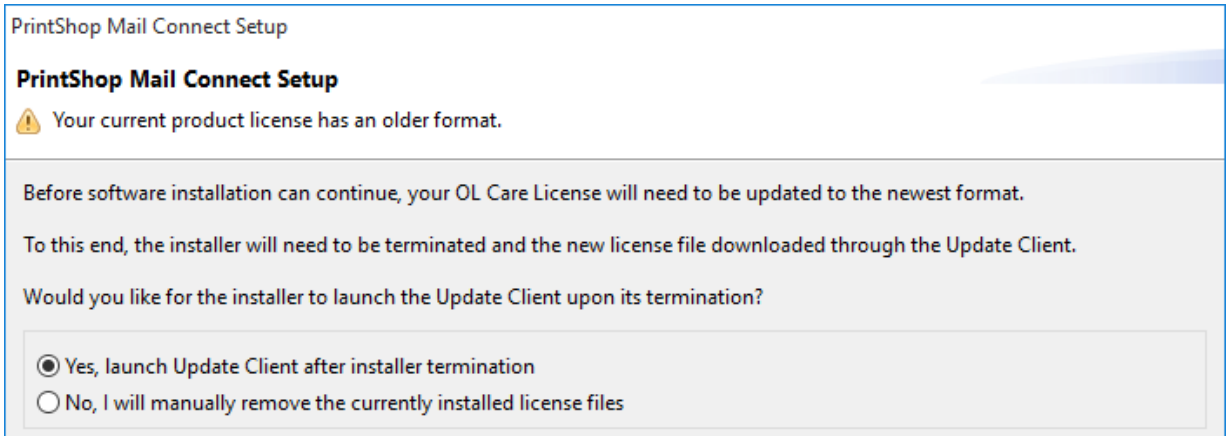
License Update Required for Upgrade to Connect 2020.x

From OL Connect 2019.1 onwards, only customers with a current OL Care subscription will be able to update Connect.

Every Connect license now contains an end date that represents the last day of OL Care coverage for each customer. Upgrades and updates of Connect will be freely available up until that end date, but will not be available thereafter, unless the OL Care period is extended.

This new licensing model requires anyone upgrading to 2020.1 from Connect versions earlier than 2019.1 to first update their OL Connect License.

A dialog box appears as part of the Connect upgrade process requesting you to do this.



It is heavily recommended that you first update the OL Update Client, as version 1.2.40 of the Update Client can update your OL License to the required version, then install Connect 2020.1.

For further details on how to upgrade the Update Client and update your Connect License see the Update Client 1.2.40 Upgrade Guide (http://help.objectiflune.com/en/archive/Update_

Client_1.2.40-Upgrade_Guide.pdf).

If issues arise, or if you need to renew your OL Care subscription, please contact your local Customer Care group.

Backup before Upgrading

It is recommended that you always backup your existing Connect preferences before upgrading to a new version. This will enable you to revert back to the previous version, in a worst case scenario in which the new version introduces issues with your existing production processes. Whilst the probability of such a worst case scenario is remote, it cannot hurt to take some simple precautions, just in case.

For instructions on how to do so, please see how to backup an existing Connect installation.

Overview

This document provides an overview of the new features and enhancements in PrintShop Mail Connect 2020.1, as well as some important installation information.

Installing PrintShop Mail Connect 2022.1

- PrintShop Mail Connect is released as a 64 Bit version only.
- Full details on installing and licensing PrintShop Mail Connect can be found in the online help, which can be accessed from the software and the installer.
- Note that PrintShop Mail Connect comes with a 30 day trial license by default.

OL Connect 2020.1 Enhancements

MySQL update and improvements

The database that comes bundled with OL Connect has been updated to the latest version of MySQL (8.0.18), along with the drivers used to connect to it. (SHARED-63665/74191)

The new packaged database will only be installed with new installations. This is to avoid any chance of issues that might occur in migrating an existing database to the new version. Existing users who want to upgrade their packaged database, will have to start with a fresh database.

Other MySQL improvements

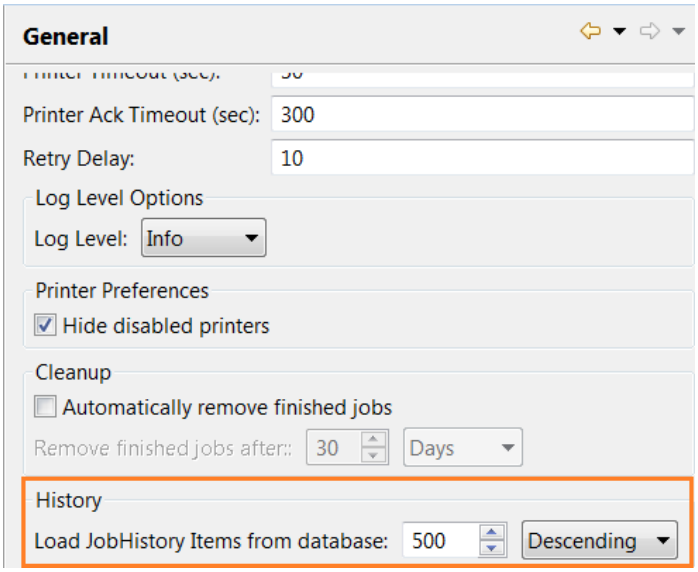
- Connect now supports connecting to existing MySQL version 8 databases for the Connect back-end database. (SHARED-71285)
- Under certain circumstances, precision was lost when using MySQL databases, causing content on pages to be misplaced. The precision of numbers has been increased. (SHARED-74458)
- Fixed issue with the Connect cleanup service failing to detect partitioning support on newer MySQL instances (v8.x). (SHARED-74623)

Security Improvements (and library updates)

- Connect dependencies have been updated to recent versions, curing a number of smaller security issues. (SHARED-73677)
- Oracle third party dependencies have been updated. (SHARED-74196)
- NodeJS version updated to version 12.13.1. All npm packages were updated to latest versions.
  Vulnerabilities listed by npm audit goes from 54 to 0. (SHARED-73468)
- A myriad of other libraries have been updated. (SHARED-74605)

Print Manager Improvements/fixes

- Speed up loading of History table, to avoid issues with cleanup services failing. (SHARED-76485/76676)
- Fixed issue with Print Manager not removing Finished jobs when the selected language was set to a language other than the default. (SHARED-76922)
- Added the ability to limit the number of jobs loaded in the job History table. The setting is available in the Print Manager General Preferences page. (SHARED-76949)



General Improvements

- Several interface translation improvements and fixes have been made. (SHARED-73925)
- Improved logging of Connect engine activity. Particularly in regards engines (data mapper, merge engine, output engine) running in tight memory conditions. (SHARED-74548)
- Externally launched engines and HSQLDB database instances will now launch with the language specified in the 'Language' preferences. (SHARED-74807)
- Fixed issue with the "background.rotation" property having effect on section clones. (SHARED-76744)

Connect 2020.1 Designer Improvements

Improved backwards compability

Save a Copy now modifies script wizards in such a way that they will work in Connect versions that pre-date the wizards.. (SHARED-74183)

Scripting Improvements

- Introduced new `translate(string[, context])` script function that can be used to translate a string and optional context. (SHARED-74827)
- **Control script change**. Assigning the value "undefined" to masterFront or masterBack should override the master page settings in the Sheet Configuration dialog and ensure that no master page is used. In 2019.2 you needed to pass either "null" or an empty string. (SHARED-74887)
- Fixed an issue with scripting that could cause statements like query("<tbody>") or query ("<tr>") to fail. (SHARED-74898)

General Designer Improvements

- The underline feature now creates wrappers with text-decoration:underline instead of <u> wrappers. This prevents a preflight warning about the <u> tag not being supported. (SHARED-69181)
- “Save as” now uses the directory of the original resource as default save location, unless the resources was never saved or was included in external package. (SHARED-70057)
- CSS can be pretty printed manually by invoking the new "**Format**" command or through the `Ctrl+Shift+F` shortcut. (SHARED-71419)
- Added support for custom properties and the `var()` keyword in CSS. (SHARED-72634) For more info, see https://developer.mozilla.org/en-US/docs/Web/CSS/--*
- Table columns can now only be re-sized in DESIGN mode. This change was introduced to prevent columns from being unintentionally re-sized in PREVIEW mode. (SHARED-73021)
- The **Styles** panel now includes styles for pseudo-elements like *::before* and *::after*. (SHARED-73327)
- Fixed a problem with the Align toolbar buttons not always properly aligning the selected elements. (SHARED-73493)

- Improved the checking of presets, to avoid prompting for unnecessary preset imports. (SHARED-74028)
- It is now possible to control which email section is processed by enabling or disabling email sections in a control script. (SHARED-74051)
- Fixed an issue with the Import Resources feature that could cause an UnsupportedOperationException error. (SHARED-74889)
- Improved the default line settings for the USPS IMb barcode. (SHARED-75097)
- Fixed an issue that would disallow changing font names in the Edit Font dialog. (SHARED-75525)

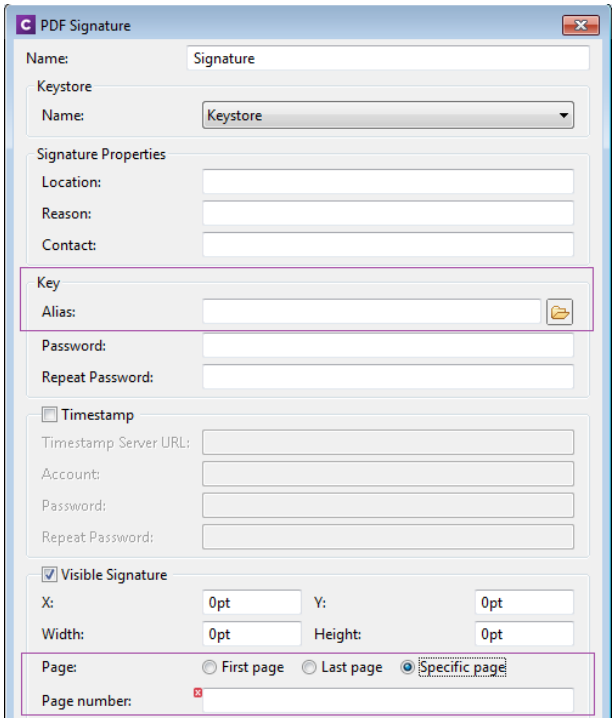
Connect 2020.1 Output Improvements

PDF Digital Signature Improvements

- The digital signature now includes the signer, the location, the reason for signing and the time of signing (if configured to use a time stamp). (SHARED-73528)
- Connect used to create digital signatures twice. Once to determine the size of the entire signature (including the timestamp if enabled) in the output file, and once for including the actual signature in the output. As of 2020.1 this "double clicking" of timestamping and digital signing has been removed, and the signature now only created the single time. (SHARED-71872)

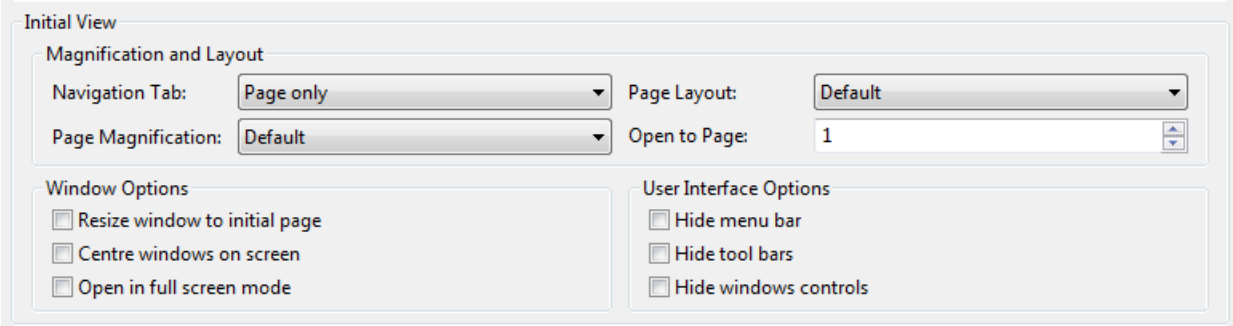
New PDF Signature dialog box options.

- Can now pick an alias from a list of aliases, using new browse button. (SHARED-75183)
- It is now possible to specify the page of the output file that the signature should appear on. (SHARED-71881)



Print Output Improvements

- A retry option has been added for when the output file exists but cannot be deleted. (SHARED-74292)
- Print Output presets now have all non-relevant entries set to default values. (SHARED-74594)
- When a table was split across more than two pages, some table rows would be missing from the output. This issue has now been fixed. (SHARED-74903)
- Fixed a bug where in certain specific conditions with templates that generate a lot of pages, content would be missing from the output (detail tables, batched pagination). (SHARED-74940)
- Improved handling of angled and skewed text in AFPDS , IPDS and PCL outputs. (SHARED-75094)
- PDF initial view options are now available as settings. These allow you to specify how the PDF output is to be viewed upon opening. (SHARED-75224)



- Fixed a crash when uncoloured tiling patterns were used with an Indexed colour space. (SHARED-75250)
- Fixed issue with TTF outline characters appearing squashed in AFPDS and IPDS outputs. (SHARED-75462)
- Fixed an issue where setting Rasterize Options on a `div` would cause errors while generating output. (SHARED-75577)
- In PostScript level 3 output, when a shading refers to a named colorspace resource, Connect now ensures that the colorspace gets written before the shading itself. (SHARED-76342)
- Fixed a problem with decimal number precision when writing out PDF Type 3 font in PDF pass-through mode. (SHARED-76406)
- Fixed an issue with Ledger paper size in PCL output. (SHARED-76630)
- PCL custom page sizes failed to select the correct size, when paper in Landscape mode. This has now been corrected. (SHARED-76634)

Email Output Improvements

- Added a new CSS inline mode for email. (SHARED-74350)
- Restored support for email attachment paths containing spaces (which was broken in 2019.2). (SHARED-75050)
- Fixed a problem with scripted email headers that was introduced in 2019.2, whereby the headers were not reset after each record. (SHARED-75413)

General Output Improvements

- Fixed an issue where in rare cases the wrong indexToLocFormat could be written in a TrueType font's 'head' table, effectively corrupting the font. (SHARED-74383)

- Fixed an issue that crashed the output engine when processing some PDF input files that containing lots of resources (such as color spaces). (SHARED-74655)
- The Dutch Postal Service PostNL KIX database has been updated to the 2020 version. (SHARED-74801)
- Improved Connect support for Type 1 fonts. (SHARED-74910)
- Fixed an issue that broke WYSIWYG output when using a resizable table in combination with the rowspan attribute. (SHARED74915)
- Fixed an issue with text positioning encountered when writing text scaled down to zero. (SHARED-75096)

General Print Wizard Improvements

- Fixed an issue where occasionally custom OMR size validation code in the Wizard would fail with an unexplained validation error, preventing the Wizard from being finished. (SHARED-74264)
- Fixed logic with media selection size rules when the media is landscape orientation. (SHARED-74540)

Known Issues

As of PrintShop Mail Connect 2018.1 the Known Issues have moved from the Release Notes to the Connect online Help.

They can be found here.

OL PrintShop Mail Connect Release Notes 2019.2

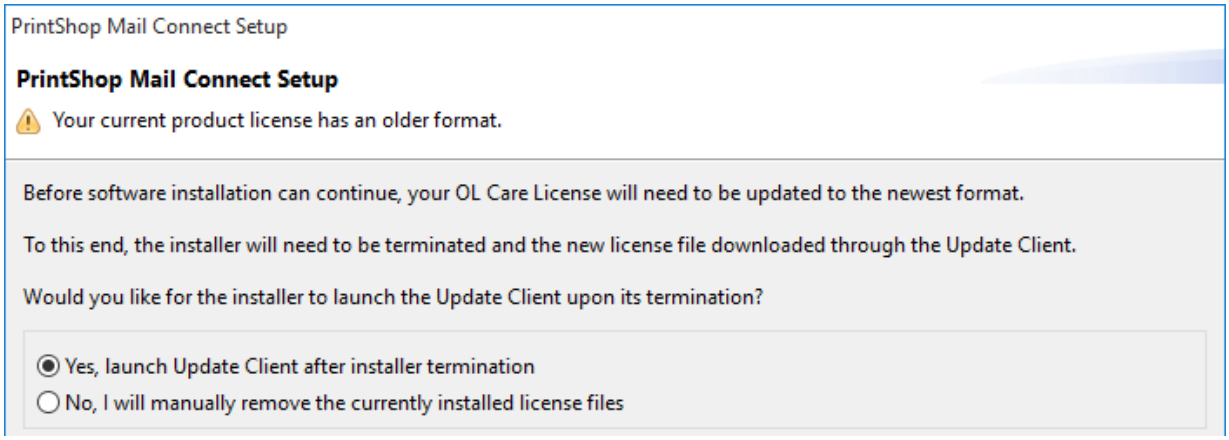
License Update Required for Upgrade to Connect 2019.x

From OL Connect 2019.1 onwards, only customers with a current OL Care subscription will be able to update Connect.

Every Connect license will now contain an end date that represents the last day of OL Care coverage for each customer. Upgrades and updates of Connect will be freely available up until that end date, but will not be available thereafter, unless the OL Care period is extended.

This new licensing model requires anyone upgrading to 2019.2 from Connect versions pre-2019.1 to first update their OL Connect License.

A dialog box appears as part of the Connect upgrade process requesting you to do this.



It is heavily recommended that you first update the OL Update Client, as version 1.2.40 of the Update Client can update your OL License to the required version, then install Connect 2019.2.

For further details on how to upgrade the Update Client and update your Connect License see the Update Client 1.2.40 Upgrade Guide (http://help.objectiflune.com/en/archive/Update_Client_1.2.40-Upgrade_Guide.pdf).

If issues arise, or if you need to renew your OL Care subscription, please contact your local Customer Care group.

Backup before Upgrading

It is recommended that you always backup your existing Connect preferences before upgrading to a new version. This will enable you to revert back to the previous version, in a worst case scenario in which the new version introduces issues with your existing production processes. Whilst the probability of such a worst case scenario is remote, it cannot hurt to take some simple precautions, just in case.

For instructions on how to do so, please see how to backup an existing Connect installation.

Overview

This document provides an overview of the new features and enhancements in PrintShop Mail Connect 2019.2, as well as some important installation information.

Installing PrintShop Mail Connect 2019.2

- PrintShop Mail Connect is released as a 64 Bit version only.
- Full details on installing and licensing PrintShop Mail Connect can be found in the online help, which can be accessed from the software and the installer.
- Note that PrintShop Mail Connect comes with a 30 day trial license by default.

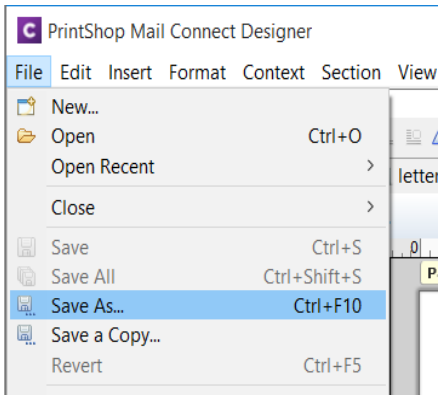
OL Connect 2019.2 Enhancements

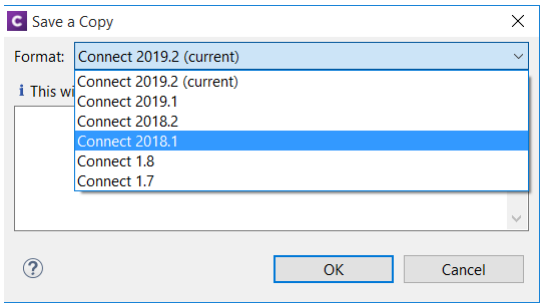
Windows Server 2019 now officially supported

As of PrintShop Mail Connect 2019.2, Connect is now officially supported under Windows Server 2019. (SHARED-69995)

Saving Connect files for different versions

You can now save Connect templates configuration files in previous Connect version formats. (SHARED-71187)



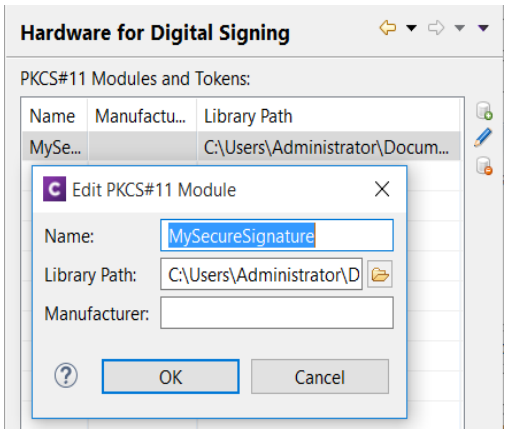


Improved PDF Signing

We have overhauled OL Connect's functionality for digitally signing PDFs. Amongst other improvements, Connect now supports signing certificates on secure devices such as USB tokens, and HSM's. This enables applying digital signatures for use cases where regulations require secure storage of the certificate. (SHARED-71456)

The PDF signing improvements include:

- Support hardware security devices (USB tokens, smart cards, and Hardware Security Modules) for digitally signing a PDF. The settings for this can be found in the Preferences dialog.



- Storing PDF signing passwords encrypted in output preset files.
- More reliable handling of time stamp server requests and responses.
- Can now add a timestamp to all types of digital signatures.
- Support for invisible signatures.
- Better compliance for digitally signed PDF/A-1b, PDF/A3-b, PDF/X-4 or PDF/VT-1 output.

Performance Improvements

- Improved output engine performance when processing common PDF files. (SHARED-70290)
- Improved default performance, for most users, courtesy of simplified and improved performance settings, and better initial assignment of Engines by the Connect installer.

Installer Improvements

- Issues with license importations have been fixed. (SHARED-67735)
- The Microsoft Access Database Engine has been updated to version 2016 (ACE 16.0) as the previously used engine (2010) is scheduled for end-of-life in 2020. (SHARED-68863)
- Under certain circumstances, some Connect Setups would fail when attempting to add the "*Enhance with Connect*" association with PDF files. This would then cause the setup to appear to fail. This issue has been fixed. (SHARED-69559)

Barcode Improvements

- Added support for the **Royal Mail Mailmark 4-State C** and **Mailmark 4-State L** barcodes. (SHARED-63467/69719)
- Several barcode names have been changed to better match how they are referred to in industry. (SHARED-72618)
  The biggest changes are:
  - OneCode has been renamed to USPS IMb
  - IMPB has been renamed to USPS IMpb
  - Royal Mail Mailmark has been renamed to Royal Mail 2D Mailmark
  - Royal Mail 4 State (RM4SCC) has been renamed to Royal Mail 4-state (CBC)

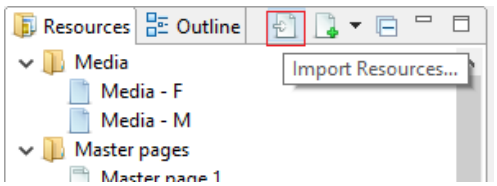
General Improvements

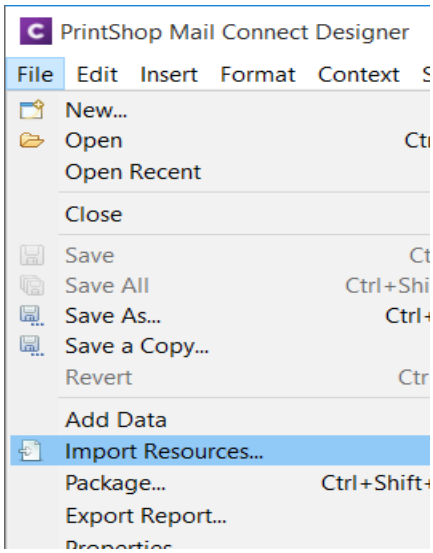
- Extra information has been added to the log files to aid in diagnosing problems. This information includes the Operating System, Regional settings as well as the available system memory and Hard Disk Drive space. (SHARED-70542/70729)

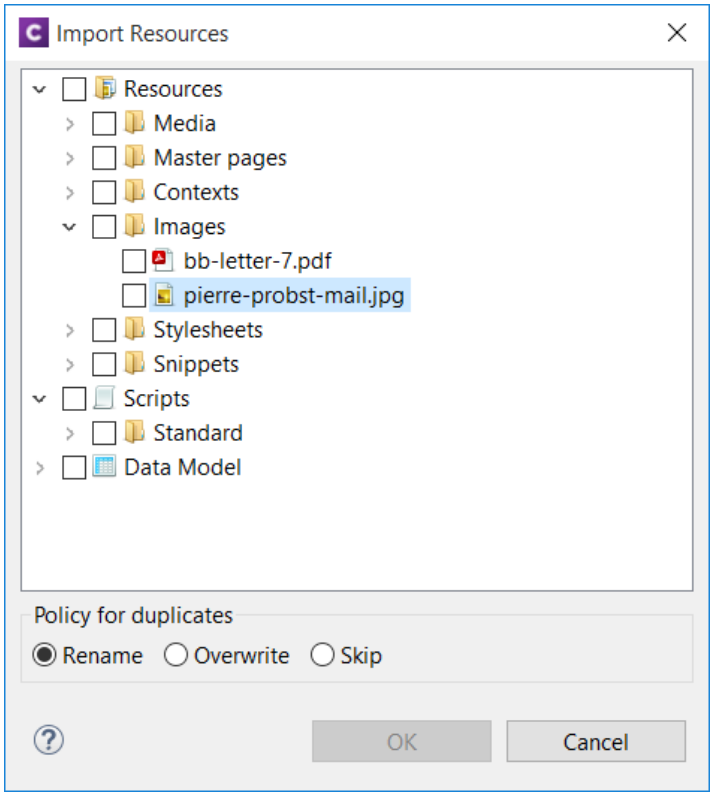
Connect 2019.2 Designer Improvements

Import resources from other Connect jobs

Introduced an option to import resources (such as scripts and snippets) and data models from other Connect template files. This makes sharing existing Connect resources much easier. (SHARED-26597)

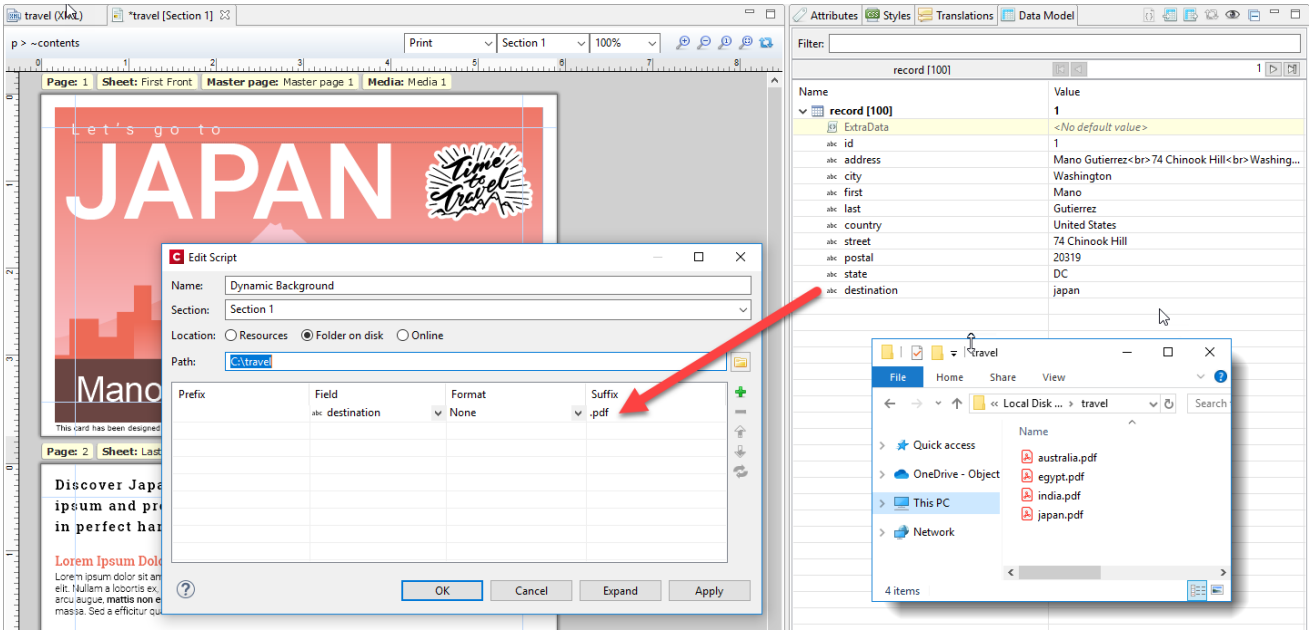






Simplified use of dynamic background images

A new scripting wizard has been introduced, simplifying creation of documents with dynamic background images. These images could be stored in the template, on a local folder on disk or remotely on a web server. (SHARED-67948/71096)



Improved white-space behaviour

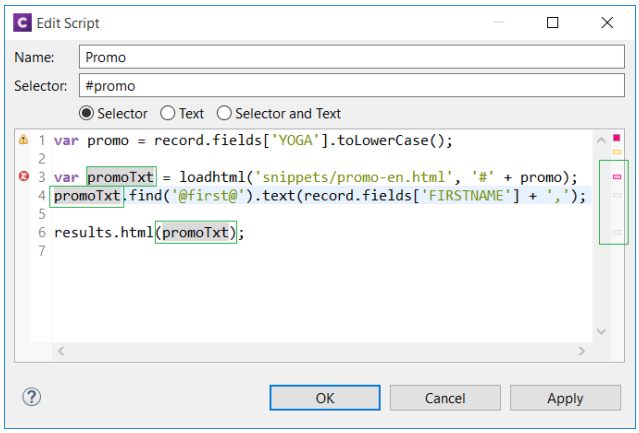
Designer adds an empty paragraph to each HTML editor, to allow easy (keyboard) editing. When this empty paragraph remains in a (non-remote) snippet, this can lead to undesired white-space in the section in which it gets included.

Designer now removes the empty paragraph from inserted HTML when the snippet is included in a section. (SHARED-68793)

NOTE: To prevent the removal of desired white-space lines (please do consider using styles instead), add a non-breakable space, or some other invisible code point.

Scripting Improvements

- When a master page editor is active in the Designer, the Post Pagination folder in the Designer Scripts panel is now decorated with an icon and a tooltip to clarify that post pagination scripts are not active in a master page editor. (SHARED-68543)
- Scripts can now differentiate between master pages and sections through an attribute selector. (SHARED-69811)
- Matches from cross-section queries are now highlighted when the mouse is hovering over a post pagination script. (SHARED-70156)
- Dragging and dropping a JSON or HTML snippet in the script editor automatically inserts the accompanying script commands (`loadhtml()` or `loadjson()`) and sets the path for the dropped resource.
  The `loadhtml()` and `loadjson()` functions now also have improved support for code completion. (SHARED-70687)
- The script editor and debugger now automatically mark all occurrences of a selected identifier, as well as marking whereabouts within the script they occur in. (SHARED-71453)



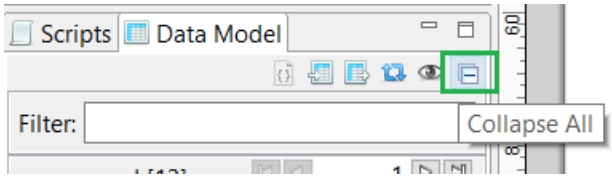
Script Debugger Improvements

- New hot keys added to the "*Find*" control in the Script debugger. They are **Enter** for "*Find Next*", and **Shift+Enter** for "*Find Previous*". (SHARED-69422)
- Improvements made to the information control that is displayed when hovering over a variable whilst a script is suspended. (SHARED-69861)
- The hover control in the script debugger now allows you to navigate to the children/parent of a results object. (SHARED-69949)
- Improved tooltips in variables panel. (SHARED-69986)

- Pressing **Ctrl+F** in the script debugger now starts an "*Incremental Find*" operation. (SHARED-70147)
- Fixed an issue whereby custom expressions were not always refreshed when stepping through the code. (SHARED-71276)
- Some cosmetic improvements made in the scripts overview within script debugger. (SHARED-72034)

General Designer Improvements

- To prevent issues it is no longer possible to replace absolute positioned direct child elements with inline elements in the body of Master Pages. (SHARED-68791)
- Formatting contained within a `<pre>` tag is no longer pretty-printed in the HTML source editor. (SHARED-69407)
- To prevent issues we no longer allow drag-drop operations in the outline tree if either the **Preview** or the **Live** tab is active. (SHARED-69640)
- We now generate a runtime error if a script tries to apply an unknown media or master name. A scripted master name with a NULL value or an empty string now resolves as "don't use a master". (SHARED-69683)
- When a standalone HTML file is opened in Designer, we now hide or disable the menu options that are not appropriate for standalone HTML files. (SHARED-69748)
- Added new `html(value)` function for Control Scripts that can be used to set the initial HTML of a section or master page. (SHARED-69782)
- Designer can now send more than 200 Emails in a single batch. (SHARED-70188)
- A **Collapse All** button has been added to the Data Model view toolbar. (SHARED-70451)



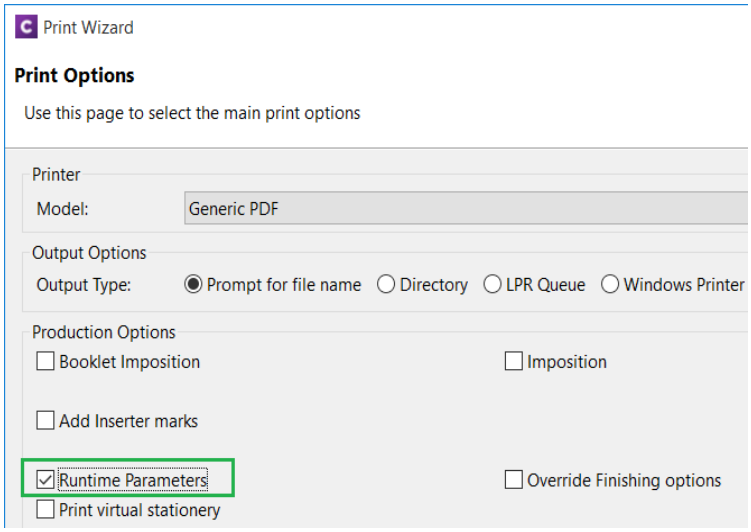
- Image file size restrictions removed. The only limit to images now is the memory available to Designer. (SHARED-70565)
- If a template is linked to a data model, code completion for `record.fields` and `record.tables` will now offer proposals even if no DataMapper config is open. (SHARED-70924)
- Can now drag remote stylesheet resources and remote javascript resources to a section or web context in order to add them to the section or context includes. (SHARED-70937)

- The **Send to Workflow** and **Package** dialogs will now only automatically add the latest job and output preset files when there is a template open with a print context. And only if those preset files where created or modified in the current session of the designer. (SHARED-71361)
- Saving a SASS partial file now auto compiles root level .scss files. (SHARED-72224)
- Modern, Roman and Script fonts are no longer available in the Designer Font ComboBox since these fonts are no longer supported. (SHARED-72391)

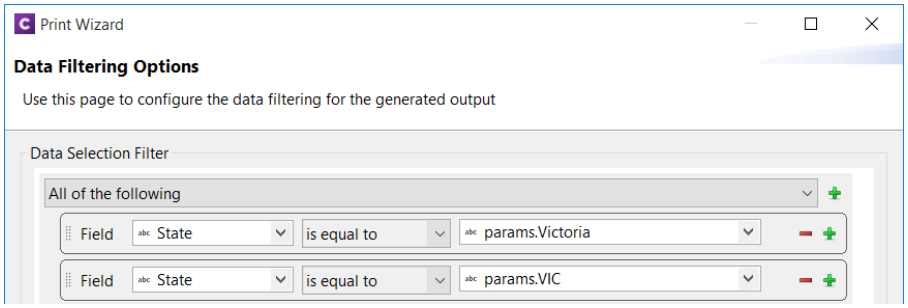
Connect 2019.2 Output Improvements

Runtime parameters added to Job Creation

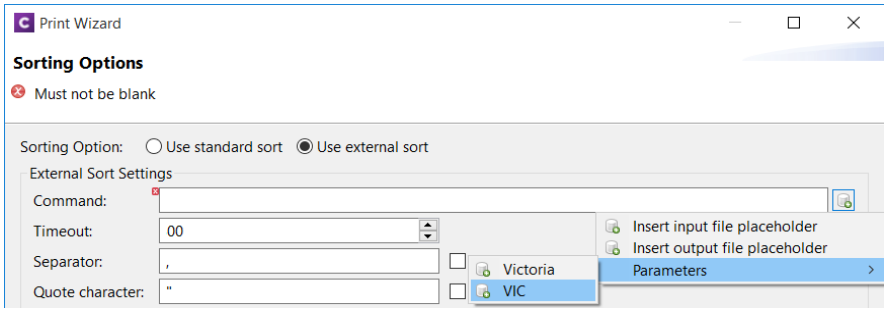
Added support for runtime parameters in Job Creation, so we can pass information from the actual runtime environment to compare against in conditions, or to use with external sorting programs. (SHARED-70085)



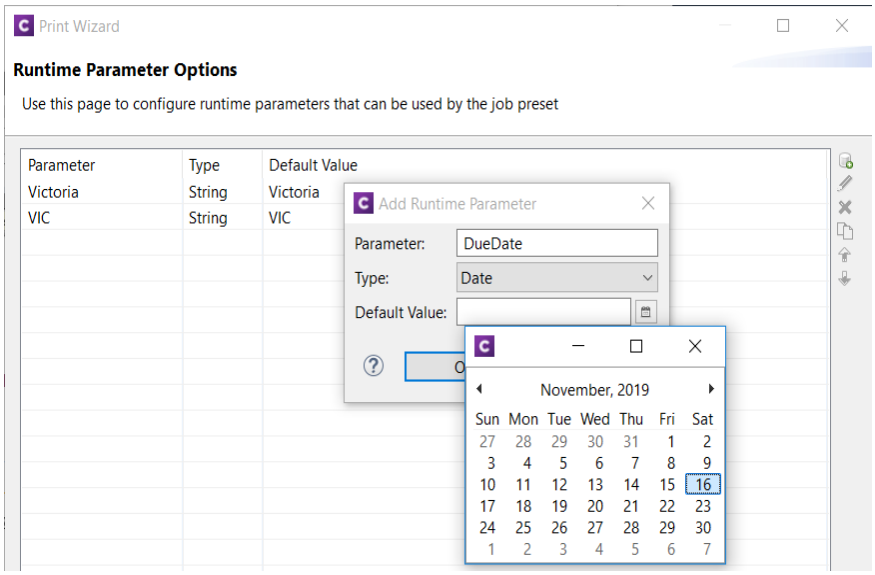
The runtime parameters will be available for use in Job Filtering and Finishing.



They can also be used as Meta Data fields, or in External Sorting.

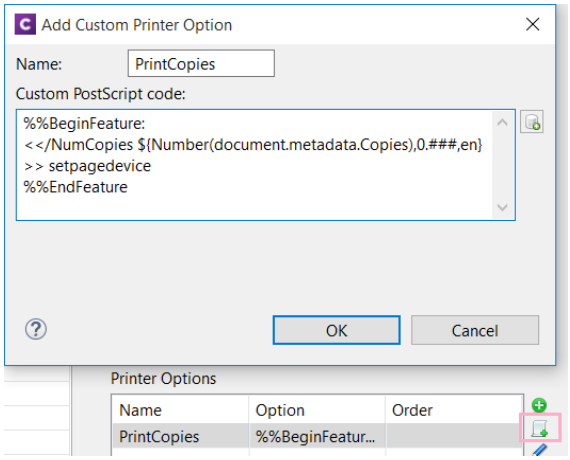


The fields can be alphanumeric, numeric, Boolean, or even dates.



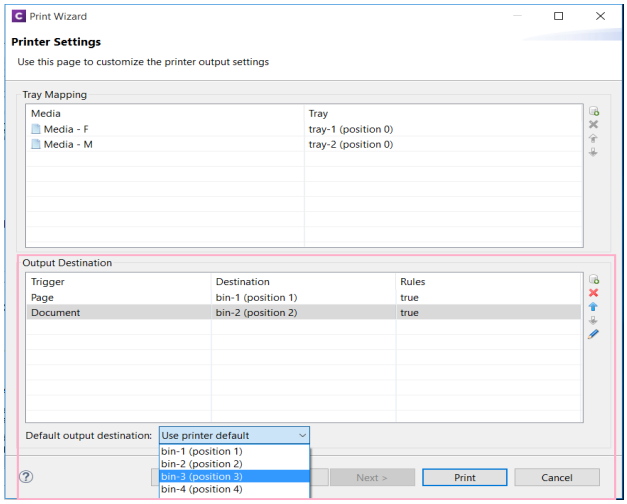
Add customized PostScript commands to Dynamic PPD PostScript output

Added support for customized PostScript commands in Dynamic PPD PostScript printer output. The commands can include metadata values, so dynamic runtime information can be incorporated within the commands. (SHARED-72170)



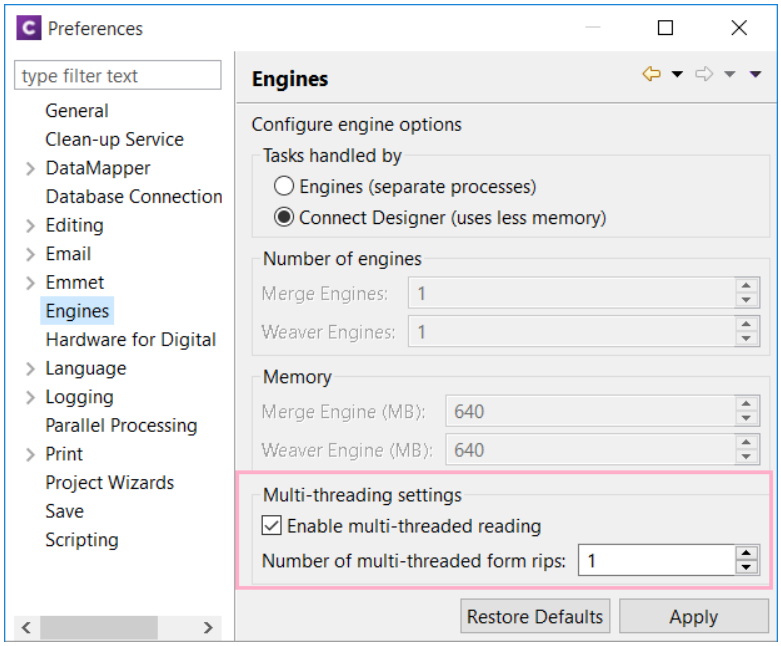
Added bin selection to AFPDS, IPDS and PCL output

Added support for bin selection in AFPDS, IPDS and certain PCL printer definitions that support bins. (SHARED-71560)



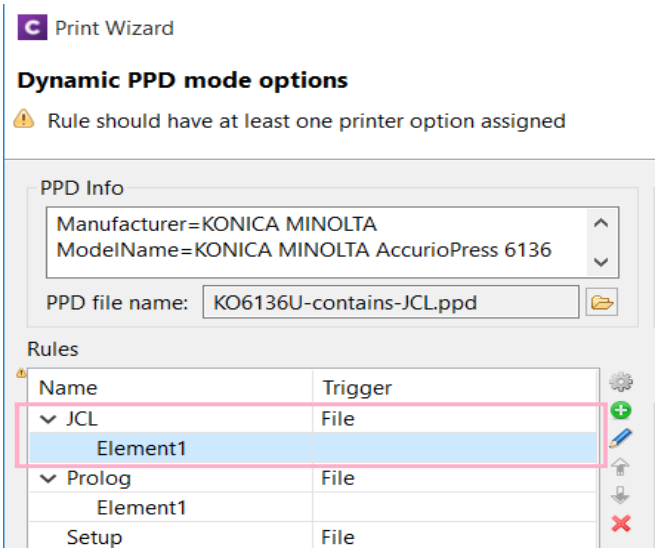
Print Output Improvements

- When using a PDF file with a specific combination of page elements as background in the template ("*use PDF from datamapper input as section background*") and creating PostScript output, it was possible that some page elements could get positioned incorrectly. This has been fixed. (SHARED-68080)
- Fixed potential PDF/A or PDF/X compliance issue when outputting multiple channels or both variants simultaneously (as some custom OL Enhance workflows do). (SHARED-69642)
- AFP: Finishing now supported at DocumentSet and Segment Level. (SHARED-70047)
- AFP / IPDS: Multi-threaded option added to improve form ripping times. (SHARED-67154) Increasing the number of threads to use with form ripping could provide speed improvements for jobs that use a lot of graphics or resources imported from other print stream files.

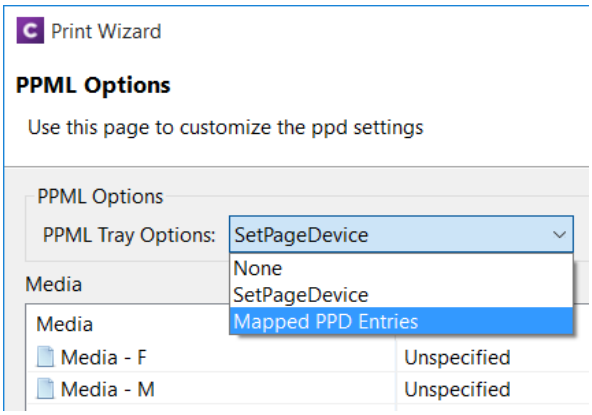


- AFP / IPDS: Native support added for RGB images in FS45. The images are no longer converted to CMYK. (SHARED-70604)
- AFP / IPDS uses an improved RIP mechanism that results in cleaner, sharper lines and improved character alignment at 300 DPI. (SHARED-70631)
- PostScript: The output engine can now write tiling and shading patterns directly instead of first rasterizing them and writing them out as images. (SHARED-65125)

- PostScript (Dynamic PPD PostScript): Enable PCL Printer Job Language (PJL) comments to be selected and added to Postscript output as Job Control Language (JCL) blocks, if supported by the printer's PPD file. (SHARED-71504)



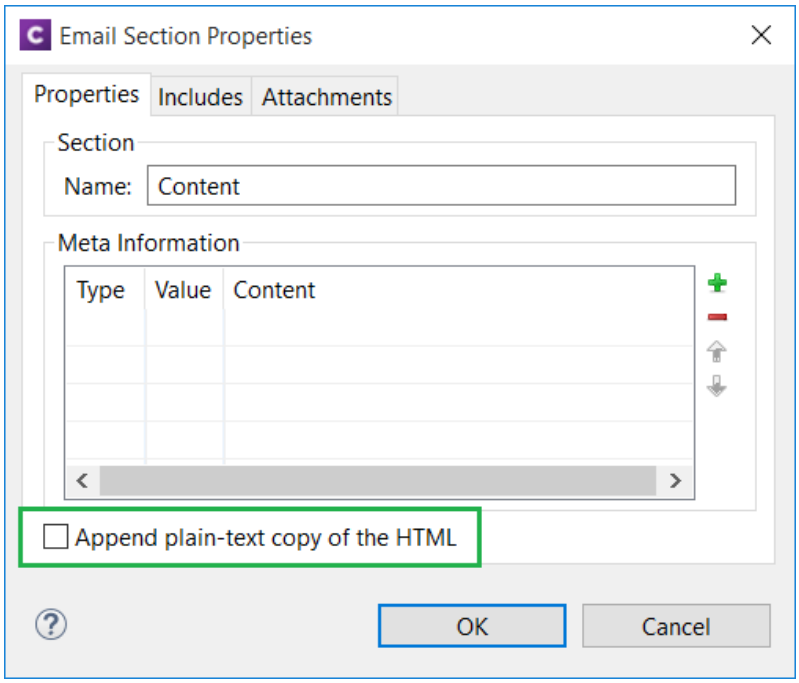
- PPML: Tray options now available for PPML print output. (SHARED-69314)



- Reduced memory usage requirements for large impositioned jobs. (SHARED-71177)
- Modern, Roman and Script fonts have been removed from the additional content font combo boxes since these fonts are no longer supported. (SHARED-72464)
- Ensured that PDF pass through is not active when the output PDF type is not standard PDF. (SHARED-72817)

Email Output Improvements

- Improved the scripting API for email headers. Connect now supports multiple headers with the same key. (SHARED-69927)
- Introduced the **Append plain-text copy of the HTML** option to email sections, to have a plain-text version of the HTML added to each email that is sent. This option is switched on by default for new templates. (SHARED-48902)



General Output Improvements

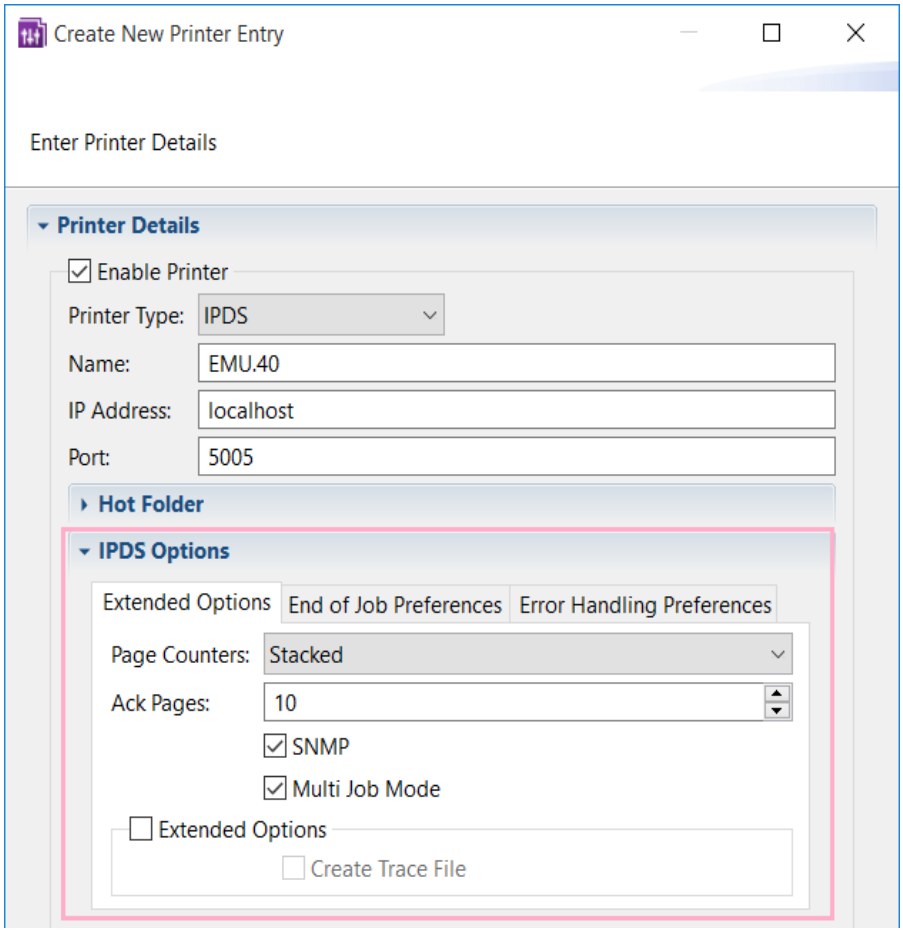
- Improved error handling when loading a template during output generation. (SHARED-67730)
- TrueType "OS/2" tables in embedded fonts are now passed through *as-is*, and the output engine will no longer abort if it does not recognize them (for example, a newer version of the format). (SHARED-69378)
- Fixed a problem with simulating font style (generating styled fonts). (SHARED-70471)

General Print Wizard Improvements

- Added support for processing tilde ('~') escape sequences in Additional Content barcodes. (SHARED-69388)
- Fixed handling of line end delimiters in Unix generated PPD files. (SHARED-70344)
- The Additional Content “EAN 128” barcode entry has been renamed to the international standard of "GS1-128". (SHARED-70798)
- Importation of PPDs improved. Some parsing issues were fixed, and the range of PPDs supported increased. (SHARED-71358/71450)

Print Manager Improvements

- Improved validation of page range values when resubmitting page ranges. (SHARED-58004)
- New **Export** and **Import** of Printer Configuration options added to Print Manager. (SHARED-59828)
- Simplified interface for setting IPDS printer options. (SHARED-72238)



Known Issues

As of PrintShop Mail Connect 2018.1 the Known Issues have moved from the Release Notes to the Connect online Help.

They can be found here.

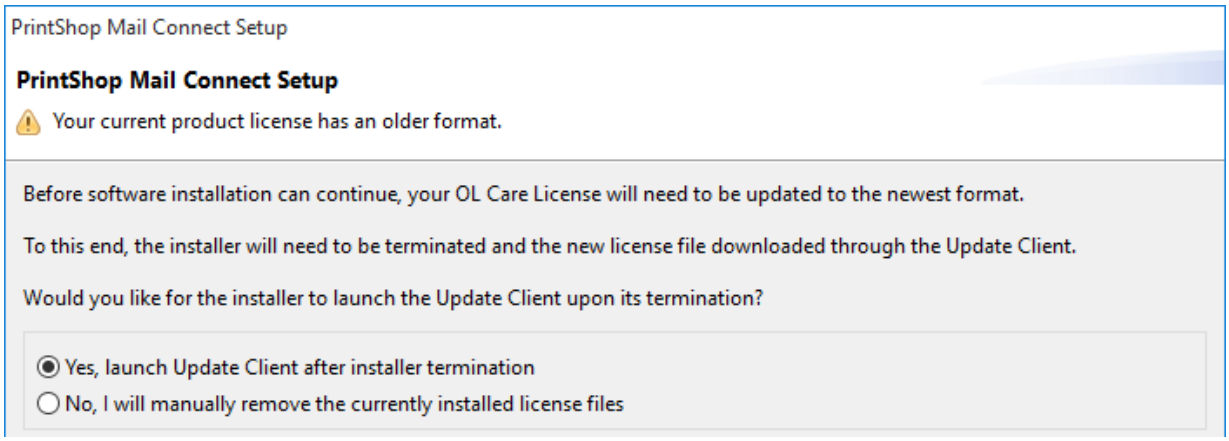
License Update Required for Upgrade to Connect 2019.1

From OL Connect 2019.1 onwards, only customers with a current OL Care subscription will be able to update Connect.

Every Connect license will now contain an end date that represents the last day of OL Care coverage for each customer. Upgrades and updates of Connect will be freely available up until that end date, but will not be available thereafter, unless the OL Care period is extended.

This new licensing model requires anyone upgrading to 2019.1 from earlier Connect versions to first update their OL Connect License.

A dialog box appears as part of the Connect upgrade process requesting you to do this.



It is heavily recommended that you first update the OL Update Client, as version 1.2.40 of the Update Client can update your OL License to the required version, then install Connect 2019.1.

For further details on how to upgrade the Update Client and update your Connect License see the Update Client 1.2.40 Upgrade Guide (http://help.objectiflune.com/en/archive/Update_Client_1.2.40-Upgrade_Guide.pdf).

If issues arise, or if you need to renew your OL Care subscription, please contact your local Customer Care group.

Overview

This document provides an overview of the new features and enhancements in PrintShop Mail Connect 2019.1, as well as some important installation information.

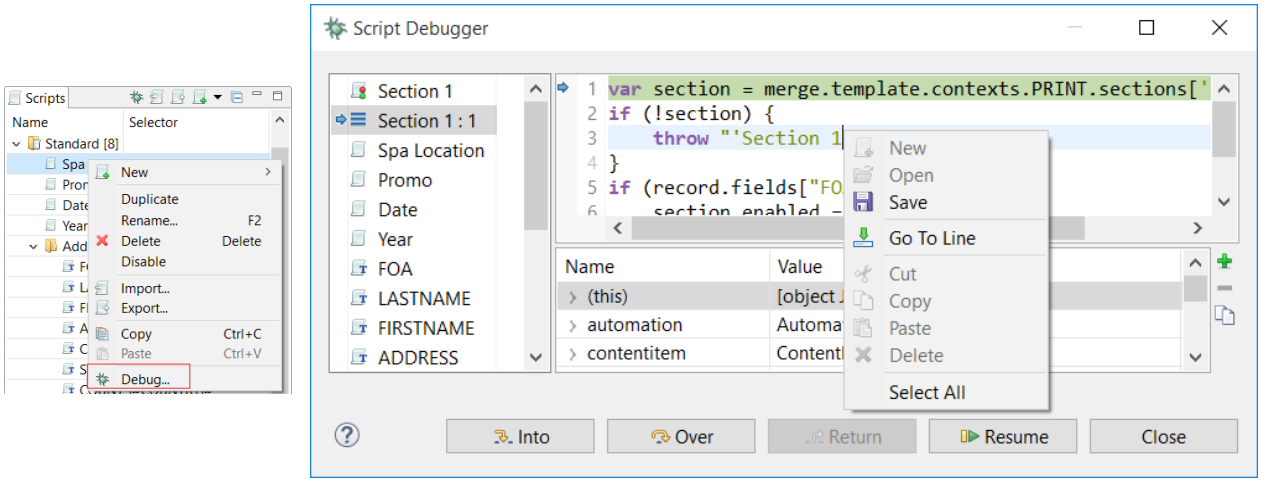
Installing PrintShop Mail Connect 2022.1

- PrintShop Mail Connect is released as a 64 Bit version only.
- Full details on installing and licensing PrintShop Mail Connect can be found in the online help, which can be accessed from the software and the installer.
- Note that PrintShop Mail Connect comes with a 30 day trial license by default.

OL Connect 2019.1 Enhancements

Script Debugger

Connect 2019.1 introduces a new **Script Debugger**. This provides options to step through code (i.e. execute the code line by line) and to add breakpoints to pause execution at strategic points, as well as displaying the current state of variables, both local and global. (SHARED-66671)



The **Script Debugger** *simulates* an output run for all sections in the current context, including section clones. The output run is limited to running scripts and pagination for the current record. A print context will not generate actual output, and an email context will not generate attachments or send the email message.

The **Scripts** pane on the left hand side of the Script Debugger lists all scripts that are enabled to apply to the current context, in the order in which they are expected to be processed.

The **Source Code** pane shows the source of the suspended or selected script. Scripts based on a script wizard show the source of the expanded script. The current line in the script is highlighted, and the left margin of the source editor has a “pointer” marking the current line position within the greater script. You can ad breakpoints by clicking in the left margin of the source editor. The script will then pause at this line.

The **Variables** pane shows a hierarchical overview of the state of all local and global variables. It includes the special entry “(this)”, which represents the JavaScript “this” object.

Improved Data Model view

- **Ordering items**

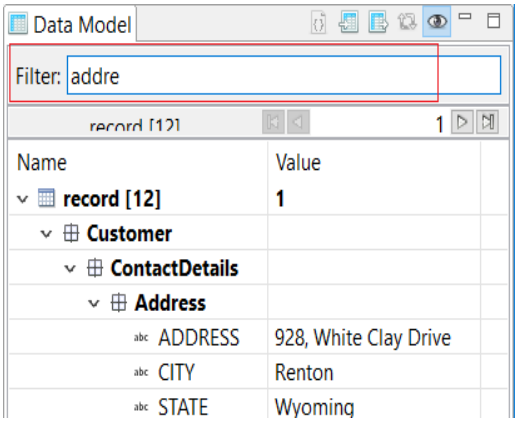
  Data Model fields can now be re-ordered to create a fully customized Data Model view. Multiple fields can be selected together and drag & dropped either higher or lower in the Data Model pane, regardless of the order in which the data is actually being extracted. This does not affect the data at all, just the display of the data.

- **Ordering selections made permanent**

  Ordering information is saved. When one opens an OL Connect Template, the Data Model view will display the same Ordering information as was last saved.

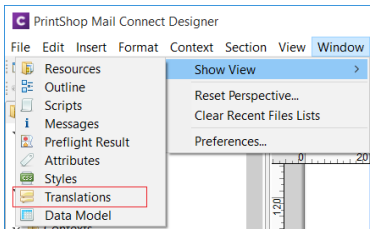
- **Filtering the view**

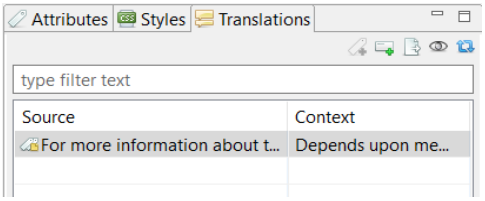
  Data Model pane now allows filtering the displayed database items via a new **Filter** box. This narrows the list of displayed items to just those tables or fields whose name contains the typed string.



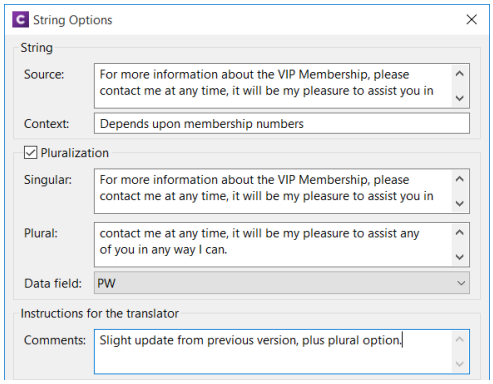
New Translation feature

A multilingual Connect template can handle content and static text elements in more than one language. Traditionally this was done through creating a separate Template or Section for each of the language variants. A new **Translation** feature in Connect 2019.1 simplifies this process and introduces options to export translation strings in a standardized file format. (SHARED-21729)





OL Connect Designer lets you import translated string files and applies the translations based upon the current locale. The locale can be explicitly set for the template or dynamically defined by the value of a data field. The Translation feature even handles pluralization of strings.



The new translation workflow in a nutshell:

- **Tag page elements** for translation (for example paragraphs, headings, table cells, buttons, labels etc.). This adds the respective strings to the *Translations* view.
- **Export** the strings to a Portable Object Template (.pot) file. Send this file to translators.
- Translators **translate strings** and send the results back as a Portable Object (.po) file.
- **Add the .po file** to the *Translations* folder located in the *Resources* panel.
- **Set the desired locale** for the template either to a specific locale or set it dynamically based upon a data field.

Installer Improvements

- You can now modify your installation choices, when upgrading Connect from an earlier version. (SHARED-67956)
- The Database Connection Check now displays a progress monitor, which is useful when attempting to connect to remote databases. (SHARED-23309)
- DataMapper engines no longer get installed in PrintShop Mail Connect installations. (SHARED-63447)

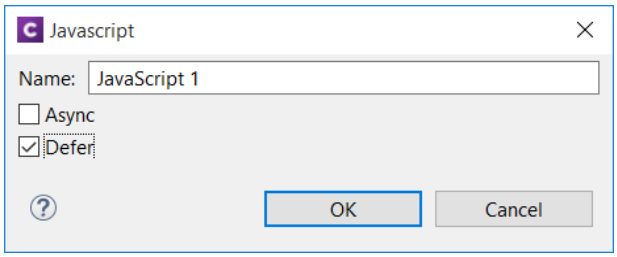
Print Manager Improvements

- Improved paging speeds when viewing/processing large print spool files. (SHARED-67984)
- Fixed issues whereby some IPDS and PCL files not being displayed entirely correctly. (SHARED-54864)

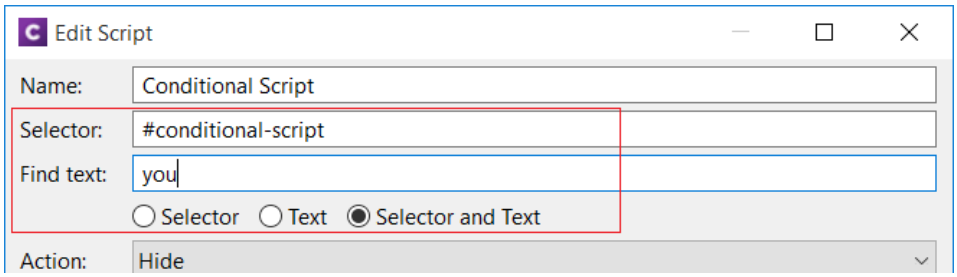
Connect 2019.1 Designer Updates

Scripting Improvements

- Regular JavaScript resources can now have *Defer* and *Async* properties set, as was already the case for remote JavaScript resources. (SHARED-67235)



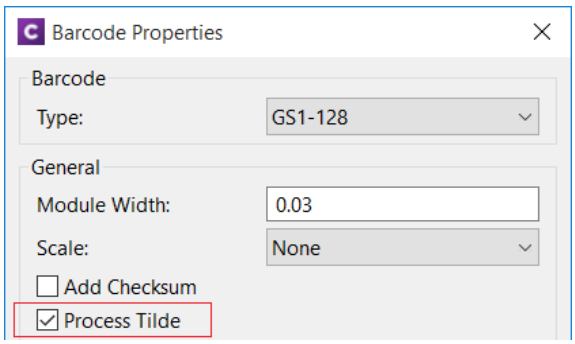
- Scripts created by the *Make Conditional* option previously just used the ID and class of an element by default for the selector, but this can now be changed as desired from within the **Edit Script** dialog. (SHARED-68412)



- Text script wizards now validate whether fields exist in the active data model, and if they have the same types. The new **Refresh Types** button in the Edit Script dialog can be used to ensure that field types match the types in the active data model. (SHARED-65371)

Barcode Improvements

- Added support for the **Japan Post** barcode. (SHARED-59708 / 59711)
- Most barcodes except the 4 state barcodes (Australia Post, Japan Post, Kix and Royal Mail 4 State) and One Code now have an option to **enable\disable Tilde Processing**. (SHARED-68565)

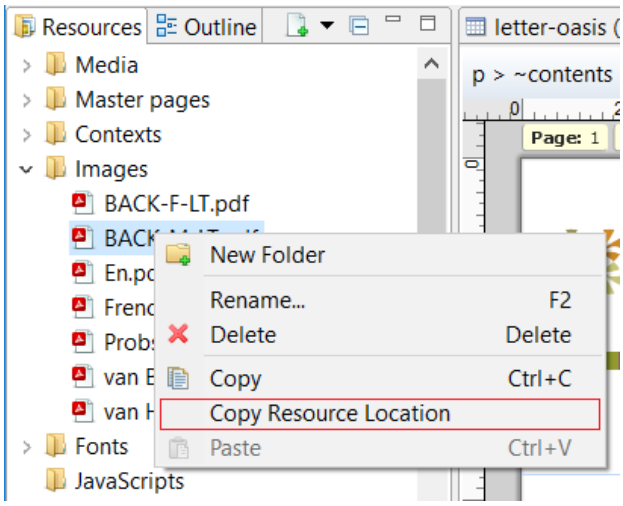


  When this option is disabled, notations like "~d009" will no longer add a tab character to the barcode value. Rather the literal value "~d009" will be encoded.

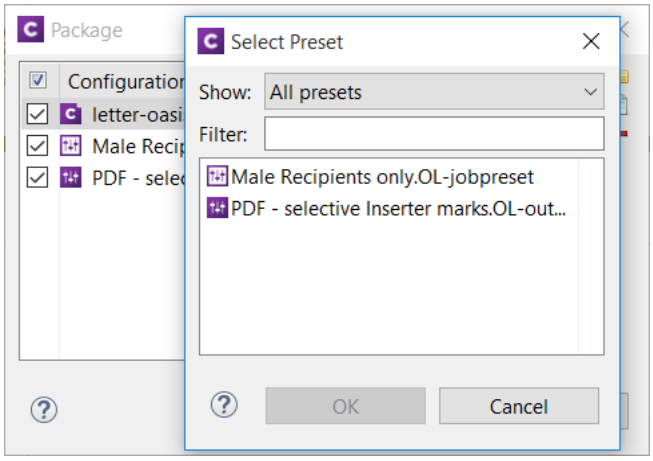
  When this option is enabled you can escape a tilde by using another tilde (~~) for those data items which contain tildes that are meant to be used literally.
- The **KIX database** of the Dutch Postal Service PostNL has been updated to the 2019 version. (SHARED-68210)
- Fixed an issue whereby the barcode properties dialog would not appear for barcodes that didn't have an ID. (SHARED-50762)

General Designer Improvements and Fixes

- Added ability to copy the location of snippets and images, so the locations can be pasted into scripts to simplify using those resources in scripts. (SHARED-33976)

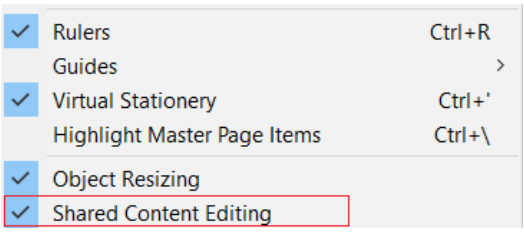


- We've added filter options to the *Package* dialog, to make it easier to locate the correct presets. (SHARED-68033)



- The *Package* dialog now automatically adds the last modified **Output Creation** and **Job Creation** configurations to the package. (SHARED-68067)
- The "Database Connection" password now only needs to be entered once on the **preferences** page. (SHARED-66660)

- Added a *Shared Content Editing* option to the *View* menu with which you can disable/enable the editing of Shared Content in the template. (SHARED-60983)



- Default section clone names weren't always made unique when creating multiple clones. This could lead to lost clones, when naming conflicts arose. This issue has now been fixed. (SHARED-64156)

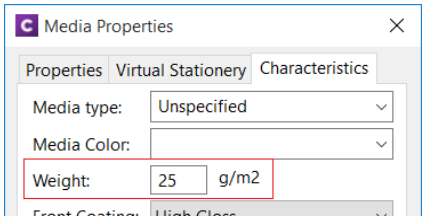
Connect 2019.1 Output updates

Font handling

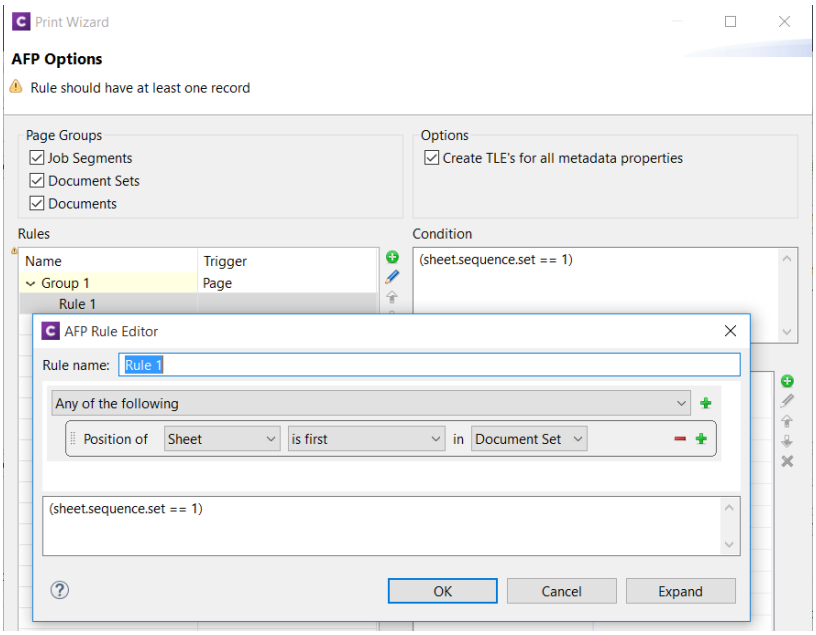
- Further improvements have been made to font handling within Connect. This should permanently resolve several of the issues previously experienced with Adobe font formats (Type 1, CFF, OpenType containing CFF). (SHARED-66561)
- Fixed problems with Type 3 font bounding boxes. (SHARED-62893)

Print Output Improvements

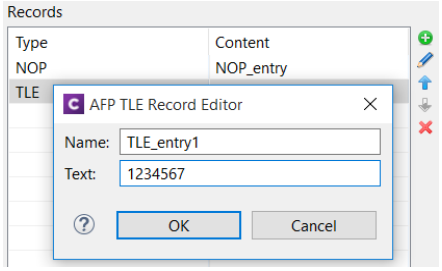
- Improved processing of Softmask graphics, to better prevent the flattening of files (which could lead to some images disappearing in printed output). (SHARED-61466)
- Enhanced print job summary in Server logs. A summary of print jobs will now be displayed in the Server logs when printing to the Server from Designer. (SHARED-66793)
- Improved page breaking. This might impact upon some existing templates. (SHARED-68453)
- Extended the range of allowable Media weights. Media weight now ranges from 25 to 1000 gsm. (SHARED-69065)



- **AFP output**
  - An AFP Rules Editor has been introduced, to support dynamically created metadata in AFP output. (SHARED-65537)

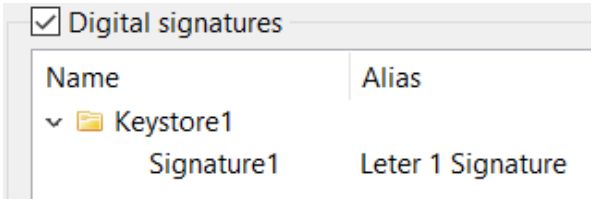


  - Added support for creating and editing NOP and TLE records, in conjunction with the new Rules option. (SHARED-65545/65548)

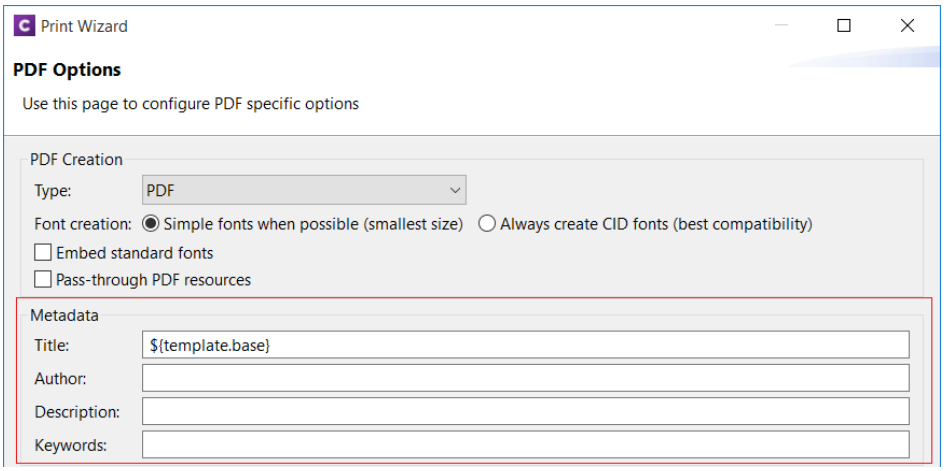


  - Connect now supports the following types of Containers: (SHARED-67629)
    - PDF and PDF Transparent
    - PNG (RGB only; supports Masks)
    - TIF-AFPC (supports Masks)
    - TIF- OPAQUE
    - TIF-TRANSPARENT

- **AFP and IPDS output** - Connect now defaults to "*Scale to Fit*" for both IPDS and AFP output. (SHARED-68389).
  NOTE: This may have an impact when printing pre-existing Connect jobs on older printer models that do not support "*Scale to Fit*".
- **PCL output** - PCL uses a newer RIP mechanism that results in cleaner, sharper and improved alignment of characters at 300 DPI. (SHARED-33148)
- **PDF output**
  - Added support for e-Invoice PDF (**PDF/A-3b**) output. (SHARED-68209)
  - Redesigned the Digital Signatures section of the PDF output options to use a tree display which better reflects the structure of digital signatures. (SHARED-68364)



  - Added support for internal hyperlinks within a PDF document. (SHARED-52674)
  - XMP data now output in Canonical format for **PDF/VT** output. (SHARED-68483)
  - It's now possible to control the metadata in PDF output. The values for the fields can be static values or dynamic values (such as values obtained from datamapping). The PDF metadata fields that can be configured are Author, Title, Description and Keywords, which are the metadata entries displayed in the Adobe Reader Document Properties dialog. (SHARED-67501/68369)



General Output Improvements

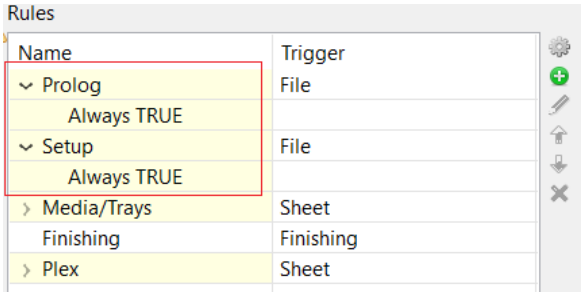
- The PDF Preview function and PDF attachments created for Email messages now have the same producer name as standard Connect PDF output. (SHARED-56002)
- Improved memory footprint of the output engine. (SHARED-67225)
- Several improvements have been made to the output engine to ensure that output creation terminates even in exceptional circumstances. For example when handling faulty input files or in tight memory situations. Further, any exception that occurs in these circumstances will be reported immediately (improving feedback on what caused the error) and output creation will be terminated as soon as possible (fail fast). (SHARED-67401)

Print Wizard and Preset Wizard Improvements

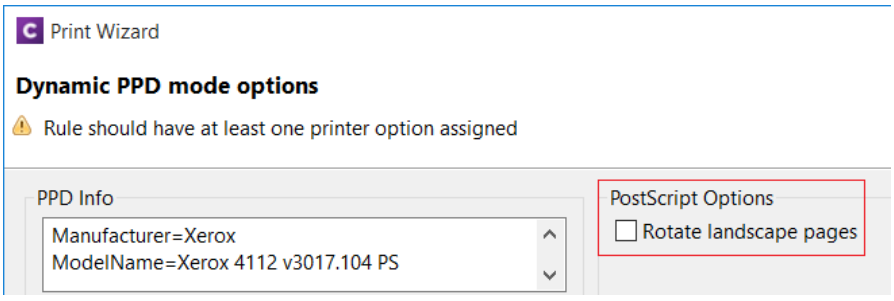
Dynamic Print Control Improvements

Improvements have been made to the creation of Dynamic Print Controls.

- When working with PPD files to dynamically drive PostScript printers, some printers have fragments called *JobPatchFile* in their PPD, which are required for the printer to function properly. These are now supported and automatically added to the output if the PPD includes such. (SHARED-68025)
- Certain printers require some options to be in very specific locations in order for them to work properly. To support this, the Dynamic PPD rules now has two groups relating specifically to two of these locations: *Prolog* and *Setup*. These rather technical features allow us to support more PostScript printers than before. (SHARED-68025)



- When adding Media in Dynamic PPD we now validate the user defined Printer Options against the Rule Details. If the rules matches a Media in the template or Presets, it is highlighted in green. (SHARED-68401)
- An option has been added to rotate landscape pages. When this option is selected, any page detected as being in Landscape orientation will be rotated to match the standard PaperSizes (which are always in Portrait). (SHARED-65191)

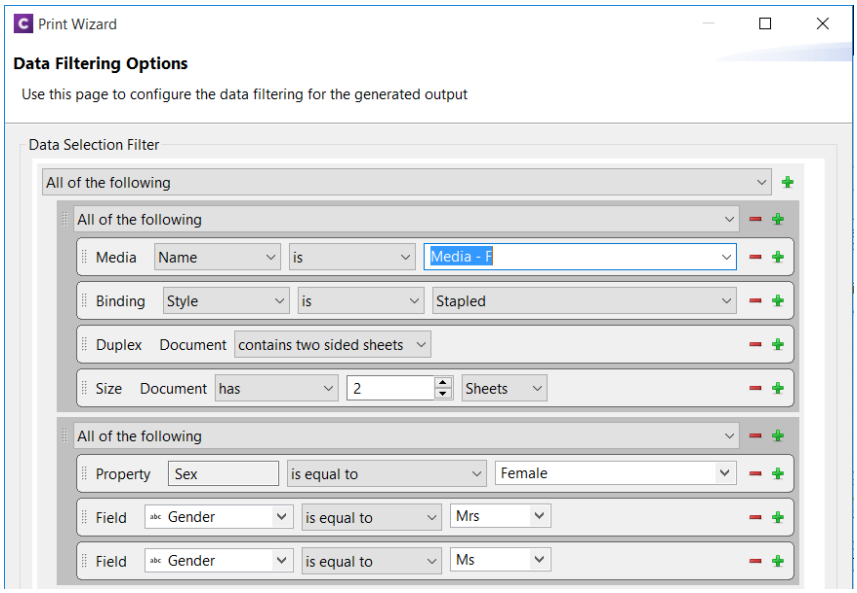


Import multiple Presets

You can now import more than one preset at a time in the Job Preset and Output Preset wizards, as well as in the Print dialog. (SHARED-66727)

Improved Data Filtering Options

The Data Filtering page in the Print Wizard has been totally revamped. It has been enhanced with extra features, and the user interface overhauled. The user interface is now much more consistent with other rule builders available within the Print Wizard.

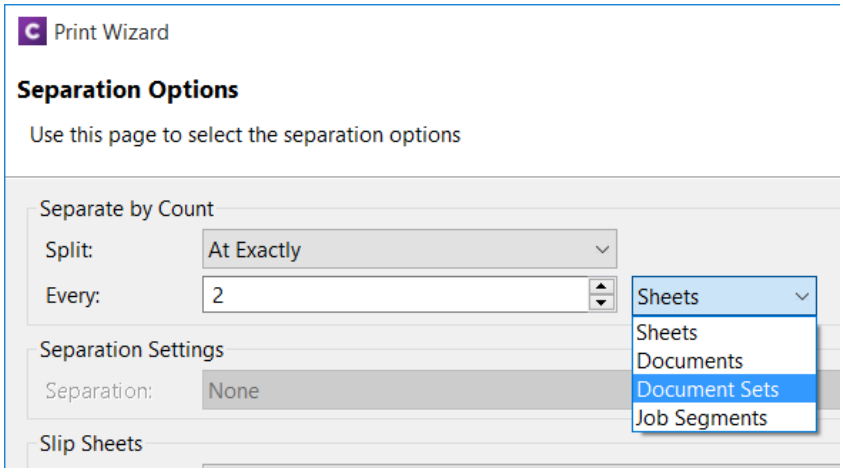


New functionality includes:

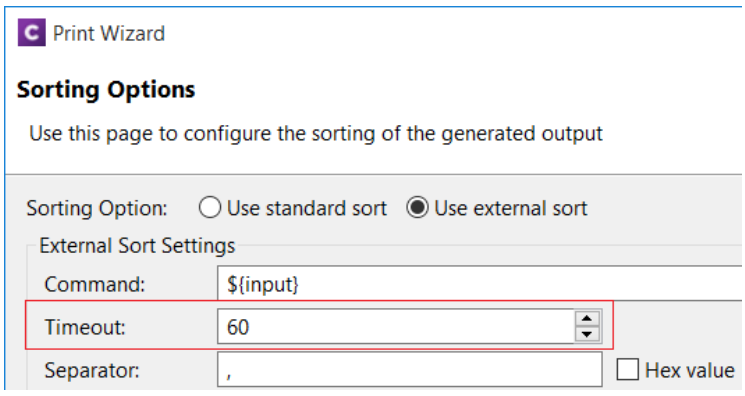
- Filter content items by number of **sheets** (rather than pages). This is better suited for filtering what fits within an envelope or inserter machine than using the number of pages,
- The use of content item **Property Rules**. This allows great flexibility because properties can be set during content creation, or from Workflow, or after data mapping has taken place.

General Print Wizard Improvements

- The **Additional Content** dialog now uses the selected Designer language's decimal and thousand separators in entry boxes. (SHARED-60881)
- Sample size preferences for standard and booklet imposition setups are now saved to the output preset. (SHARED-66484)
- In addition to being able to create multiple print files based on the number of sheets per file ("*Sheet Count Splitting*"), it's now possible to create multiple print files based on either the number of documents, documents sets or job segments per file. (SHARED-61932)



- Replaced the global timeout value (previously available for external sorting with a job preset specific timeout value. (SHARED-67982)



Known Issues

As of PrintShop Mail Connect 2018.1 the Known Issues have moved from the Release Notes to the Connect online Help.

They can be found here.

PrintShop Mail Connect Release Notes 2018.2.1

Overview

This document provides an overview of the new features and enhancements in PrintShop Mail Connect 2018.2.1 , as well as some important installation information.

Installing PrintShop Mail Connect

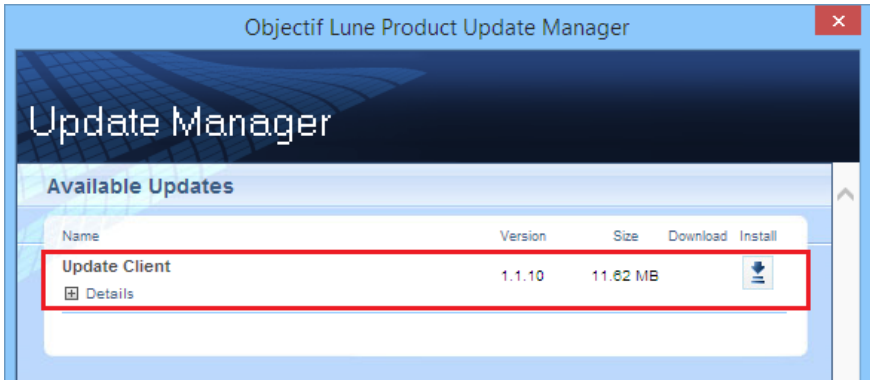
- PrintShop Mail Connect is released as a 64 Bit version only.
- Full details on installing and licensing PrintShop Mail Connect can be found in the online help, which can be accessed from the software and the installer.
- Note that PrintShop Mail Connect comes with a 30 day trial license by default.

Upgrading from PrintShop Mail Connect 1.7

It is highly recommended that you update the **Objectif Lune Update Client** before upgrading PrintShop Mail Connect from version 1.7 to version 2018.2.

If you do not update the Update Client, an unexpected error might occur whilst updating Connect. This error does ***not*** prevent the successful upgrade of Connect to 2018.2, even though it appears as if it might have. To avoid potential confusion, we recommend that you first update the **Objectif Lune Update Client** before attempting to upgrade Connect from version 1.7.

The Update Client will show that there is an update available for itself. Simply click on the download button in the dialog to install the new version of the Update Client. Note that it is no problem to run the update while the Client itself is still open. It automatically updates itself.


Objectif Lune Product Update Manager
Update Manager
Available Updates
Name
Version
Size
Download
Install
Update Client
1.1.10
11.62 MB
Details

Connect 2018.2.1 Enhancements/Fixes

Backwards Compatibility Issues

Connect 2018.2 introduced some issues which have now been addressed in 2018.2.1. These issues were:

- External sort metadata fields were not being catered for properly in the metadata validation code in the Output Preset and Print Wizards. (SHARED-67687)
- Fixed a problem affecting scripted Section configurations containing different scripted values for different records. The merge engine would sometimes use previously applied scripted values. (SHARED-67699)
- Fixed a problem with cloning template Sections that did not successfully clone all fields. (SHARED-67702)

Connect 2018.2 Enhancements

Sassy CSS

PrintShop Mail Connect 2018.2 introduces **Sass** CSS preprocessor functionality to the Designer. (SHARED-64625)

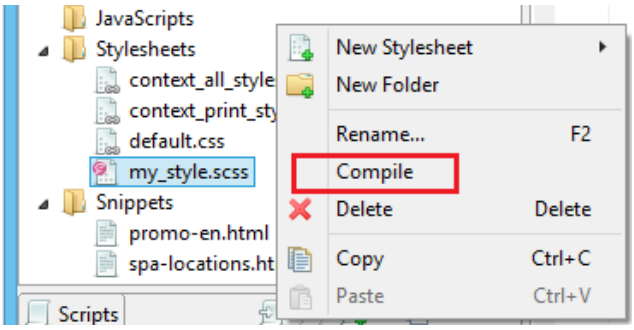
Sass (*Syntactically Awesome Style Sheets*) is an extension of CSS that enables you to use things like variables, nested rules, inline imports and more.

Maintenance of CSS styles for templates requiring many different style rules can become a tedious undertaking. Stylesheets files keep getting larger and more complex and thus harder to maintain. Some common challenges include:

- How to group/organize rules that belong together? (e.g. styles for an address block, footer or dynamic table)
- How to update a color value or font family across multiple rules and even across stylesheet files?

A CSS preprocessor like Sass solves these problems and helps you write maintainable code.

In Connect Designer options have been added to the Stylesheets folder to create and compile Sass files (.scss file written using Sass 3 syntax). For more information on the Sass language see: www.sass-lang.com.



Compiling a .scss file outputs a normal CSS stylesheet file which is automatically linked to the current section.

Alternatively use drag and drop to link the result to other sections or use the Includes dialog of a specific section.

Dynamic Sheet Configuration Settings

A common requirement for dynamic print templates is the ability to switch between simplex and duplex. Typically this is solved by duplicating a Print Section, enabling Duplex in the Sheet Configuration dialog and then switching between the simplex and duplex sections by Control Script (or through the Conditional Print Section wizard).

In Connect 2018.2 new scripting commands have been added to dynamically set Sheet Configuration options from within a Control Script. This allows you to set the Duplex mode dynamically, or do things such as applying a different Master Pages based upon the data, or omitting empty sides from the last sheet, or single sheets.

The following Control Script sample enables Duplex mode for "*Section 1*":

```
let section = merge.template.contexts.PRINT.sections["Section 1"];
section.sheetConfig.duplex = true;
```

The following snippet sets the media and the master pages for the all sheet postions of "*Section 1*":

```
let section = merge.template.contexts.PRINT.sections["Section 1"];
section.sheetConfig.positions.all.media = "My Media";
section.sheetConfig.positions.all.allowContent = AllowContent.ALL_SIDES;
section.sheetConfig.positions.all.masterFront = "My Master Page";
section.sheetConfig.positions.all.masterBack = null;
```

Performance Improvements

- **Job Creation:** Improved job creation speeds. This is particularly the case with any job creation that involves any processing, including filtering, grouping or sorting. (SHARED-50328)
- **Output:** Improved memory handling, leading to even better stability.

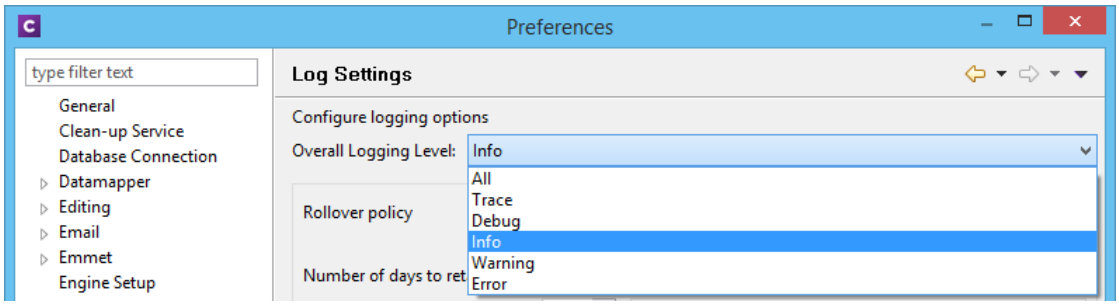
Installer Improvements

- Support added for using **non-Latin-alphabet characters** in the User Name for connecting to external databases. This applies to both the Setup and the Designer Preferences dialog. (SHARED-65243)
- **Microsoft Visual C++ Runtime** now only installed when really required. (SHARED-63909)
- Several third party libraries have been updated. (SHARED-57163)
- A warning is now displayed if the **Windows Management Instrumentation** service is not enabled and started. (SHARED-61548)
- Installer now logs more helpful messages when failing to connect to an existing MySQL server. (SHARED-62836)

Logging Improvements

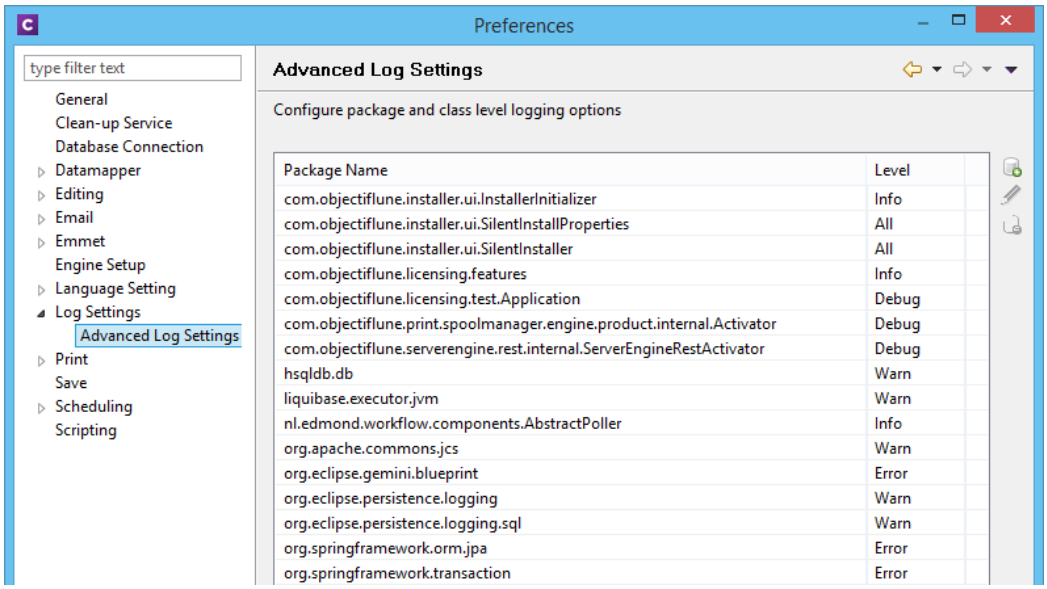
- Improved error/warning logging in Live mode. (SHARED-49046)
- Extra preferences added to allow greater control over logging levels. (SHARED-63923)

  One change is the introduction of an option to set the **Overall Logging Level**. By default the logging level is set at the midpoint of *Info*, but it can be set higher to include more logging (*All*, *Trace*, *Debug*) or lower to reduce the logging (*Warning*, *Error*).



  NOTE: Higher logging settings will have an impact upon Connect production speeds, as well as leading to substantially larger log files.

We have also introduced **Advanced Logging** preferences. The Advanced settings override the Overall logging settings, and provide a greater level of logging granularity. The Advanced Log Settings should only be set in conjunction with advice from OL support, to ensure that only the most relevant settings are set to the higher logging levels.



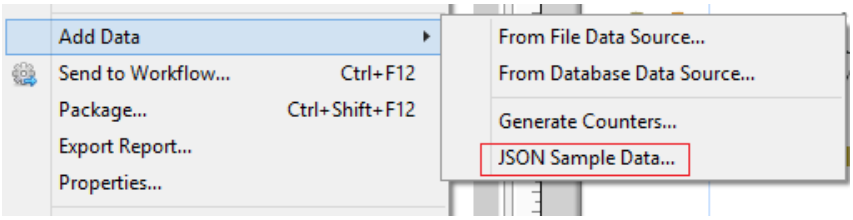
- The logging format default has changed from dd MMM yyyy (31 May 2018) to yyyy-MM-dd (2018-05-31) to make it easier to sort log files when using tools to merge and view multiple log files together. (SHARED-64233)
- Connect will now log a warning message in the event that there are insufficient engines currently available for a specific job, leading to the scheduler having to wait. (SHARED-61642)
- The location of the logging configuration file has moved from the Users directory to ProgramData to restrict the configuration to a per-machine basis. (SHARED-64147)
- Should a fatal error occur in the Java Virtual Machine running Connect the logs generated are now controlled and will appear in "*C:\ProgramData\Objectif Lune\OL Connect\ErrorLogs*" folder. (SHARED-64597)

Connect 2018.2 Designer Updates

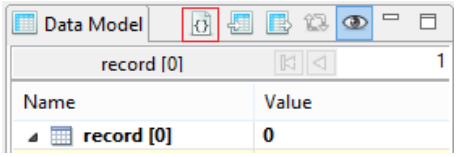
JSON support added to the Data Model Panel

The Designer now allows you to add **JSON sample data** straight into the Data Model for debugging purposes. (SHARED-61655)

This can be done via a new **File menu option**



Or from the Data Model Panel itself, using the new **JSON Sample Data** button.

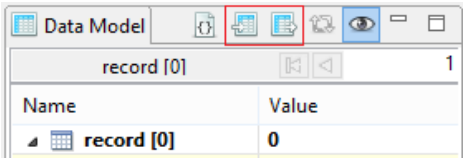


Invoking this option launches a wizard which allows you to either select a JSON file or paste JSON data copied from a Workflow variable. When loading data from a disk file the file contents are automatically placed in the JSON editor.

If JSON sample data is active invoking the **JSON Sample Data…** option will show the current data allowing you to edit its structure and values.

You can also now import or export a Data Model in JSON format. (SHARED-64620)

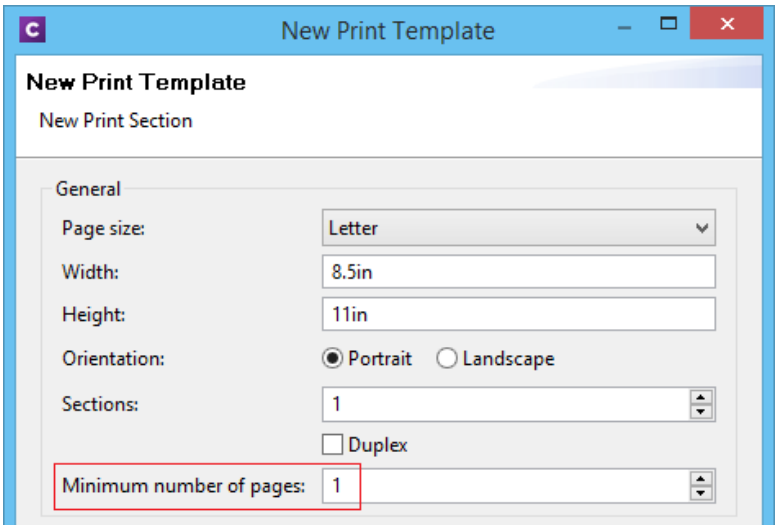
Select the Import or Export button and then choose *.JSON files type.



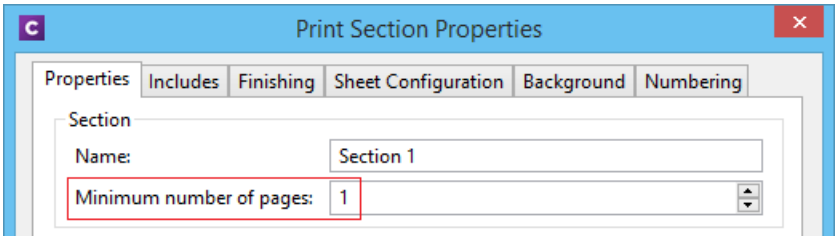
Minimum Pages option added to Print Sections

Connect 2018.2 introduces a new section property for controlling the minimum number of pages for a Print Section. This can be used to generate pages with only absolute positioned content, simplifying workflows where the number of pages are known up front and you want to draw boxes (absolute positioned divs) on specific pages. (SHARED-10236)

The property can be set when the new Template is being created.

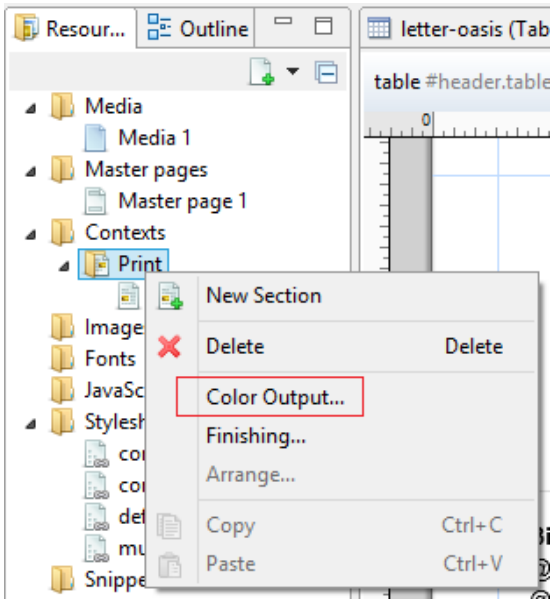


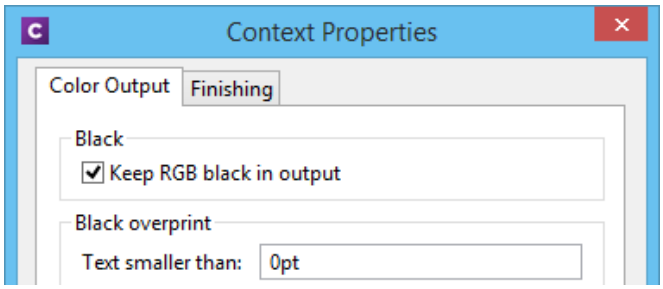
Or through the Print Section properties.



Color Output option added to Print Contexts

For certain full color print jobs, it is important to be able to set black text on top of color areas to overprint. This is of particular importance for small text, because mis-registration problems are more visible with small text. To support this a new Color Output option has been added to the Print Context. (SHARED-62265)

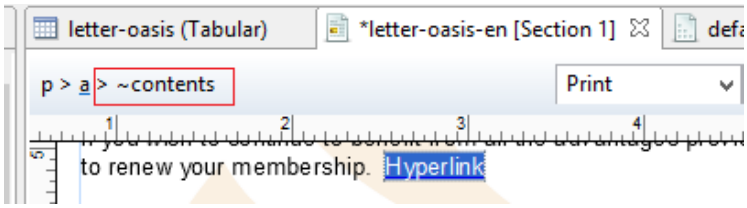




The default encoding for the color black has also changed from RGB to CMYK. Thus the "*Keep RGB black in output*" option was added for backwards compatibility.

Breadcrumb Improvements

The breadcrumbs at the top of the workspace will now always end with a ~contents breadcrumb. When clicked this will select the *contents* of the last element rather than selecting the last element. (SHARED-17339)

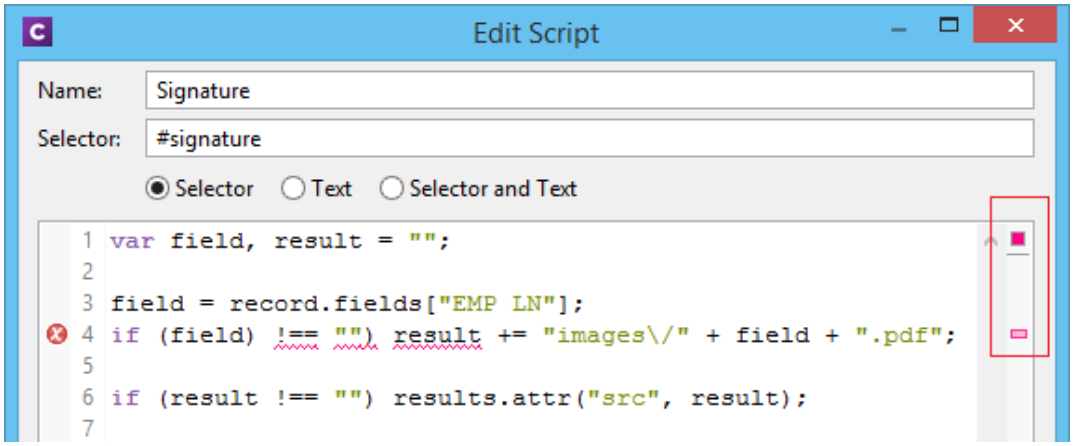


Paste as Plain Text option added

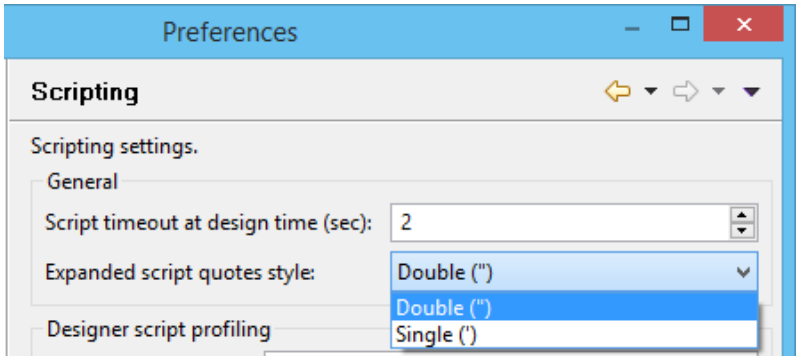
Option added to paste content copied from an external editor as plain text. This ensures any unwanted formatting and invisible elements are removed. (SHARED-63025)

Scripting Improvements

- An overview ruler has been added to the Script Editor. The ruler shows annotations concerning the entire script. These annotations are shown relative to their position in the script and do not move as the user scrolls the script source. Script errors are highlighted by a red icon, and warnings in yellow. The topmost icon will display red if any errors exist in the script at all.



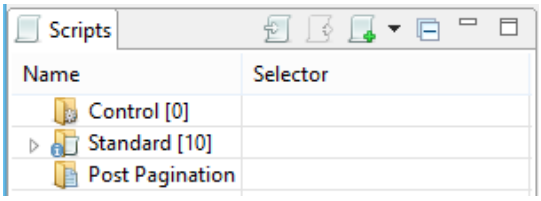
  This allows you to immediately see if a script has warnings or errors and easily jump to any warning/error location by clicking on the annotations in the overview ruler. This is particularly handy when a script has many lines of code. (SHARED-64717)
- Personalization scripts can now use the "`loadtext`" function to load text content from files. (SHARED-61512)
- Option added to Scripting Preferences to set whether expanded scripts use single or double quotes. (SHARED-60370)



- Scripts can now conditionally trigger a fatal error by calling the new `fatalError(message)` function. This will abort content creation. (SHARED-64789)
- Scripts in Print Context jobs can now attach arbitrary properties to *content items* generated by Content Creation. The properties are retained in the database, allowing the information to be utilized in Workflow for further processing along with the base record information when retrieved by the Retrieve Items action. (SHARED-65239)
- Introduced "`width`" and "`height`" script functions with the same behaviour as jQuery's "`outerWidth`" and "`outerHeight`". (SHARED-65513)
- Fixed an issue whereby the `resource()` function did not work properly when passing a UNC path. (SHARED-65523)

Post Pagination Improvements

- We have introduced a new **Post Pagination** script type, and have updated the Scripts panel so that it now groups scripts by type. The groups are Control, Standard (which contains any pre-existing scripts) and Post Pagination. (SHARED-64634)



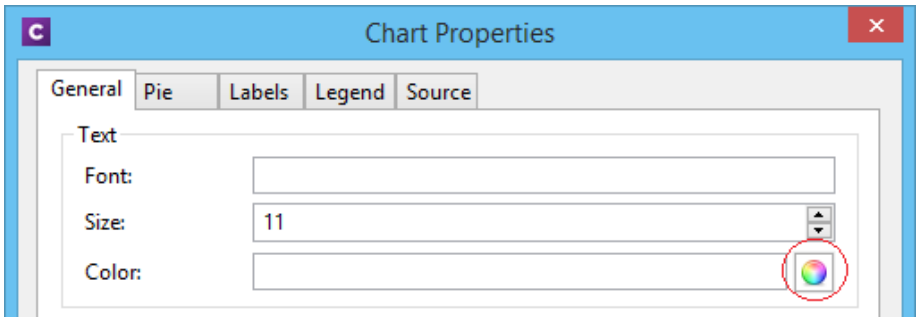
  These script types are grouped to emphasize their execution order. Control scripts are executed before the merge process and therefore run before the Standard and Pagination scripts.
- Several new script functions have been introduced for use with the new Post Pagination scripts. Scripts can now access elements in any section as well as retrieve information that is only available after pagination. This will allow for the creation of post content elements like a **Table of Contents**. (SHARED-64637)
- Margins for Sections and Master Pages are now scriptable, through the new `margins` object. (SHARED-62173)
- The sheet configuration of a Print Section is now scriptable, via the new `sheetConfig` object. (SHARED-63061)

Barcode Improvements

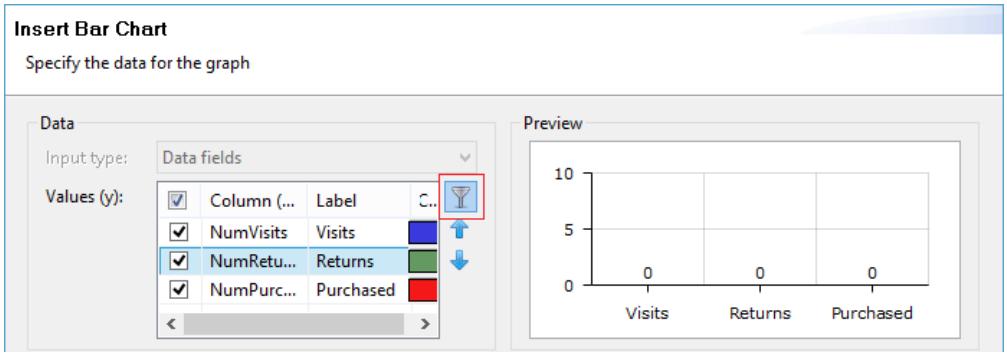
- Added **human readable text** and checksum controls to Royal Mail 4 State barcodes. (SHARED-47422)
- Added support for Grayscale “Colors”. (SHARED-50614)

Business Graphics Improvements

- **Color picker** options added throughout the Chart Properties dialog. (SHARED-62564)

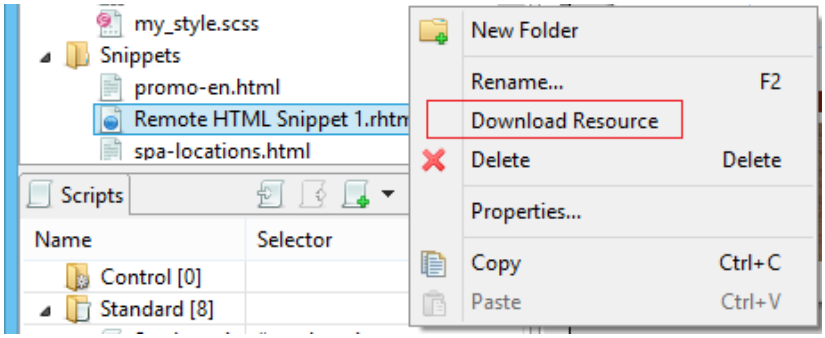


- The "*Values (y)*" table in both the **Insert Graph** Wizard and the **Graph Script** dialog can now be filtered to show only numeric fields. This will then limit the display to only Integer, Float and Currency fields in the Data Model. (SHARED-64875)

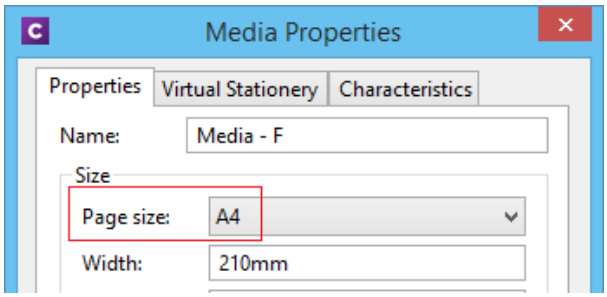


General Designer Improvements

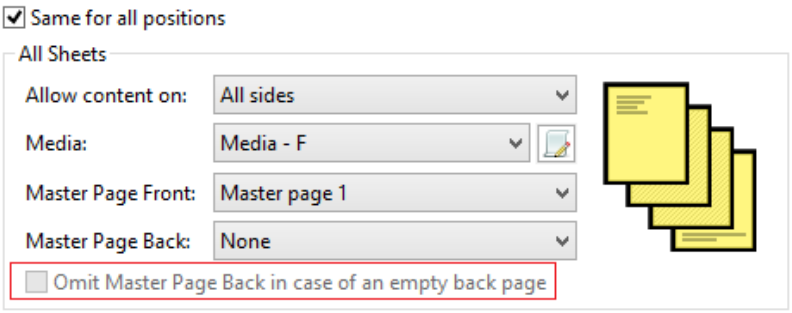
- **Download Remote Resources** (CSS stylesheets, JavaScript files, JSON and HTML snippets) to your template through a new context menu in the Resources view. This allows you to keep a link to a centralized file but quickly download a copy to your template without having to separately copy and paste.(SHARED-58949)



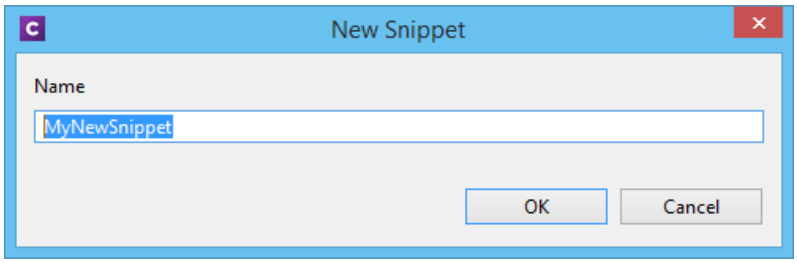
- **New Media** now defaults to the same paper size as the current Print Section. (SHARED-34012)



- Added an "**Omit Master Page Back in case of an empty back page**" option when configuring all sheets at once in the Print Section Properties. (SHARED-56383)



- **Media Properties** dialog behaviour improved. When any of the media in the Resources tree properties are changed, the "Page Setup" button gets activated and Designer prompts to save the media properties changes. (SHARED-62353)
- Disabled **irrelevant menu items** in the View menu when a Snippet editor is active. (SHARED-62816)
- When selecting content in the main editor, and selecting the context menu option of Create Shared Content or Create Snippet, you are now presented with a dialog in which to specify the name of the snippet resource. (SHARED-63217)



- Improved support for editing shorthand **!important** rules in the Edit Rule dialog. (SHARED-62863)
- **Preflight**: If a fatal error occurs during Data Mapping, the reported error will now also include the current record number. (SHARED-63089)
- **Help button** added to both the *Profile Scripts* and *Preflight* dialogs. (SHARED-63091)
- Template Wizard file comboboxes now add "Empty" value if not set. (SHARED-63294)
- Improved behaviour for when the **Auto Save** directory no longer exists. (SHARED-64428)
- Improved the way *Create Snippet* and *Create Shared Content* work. Creating a snippet for a selection of multiple elements in the Outline pane now creates a single snippet containing all the elements. (SHARED-64445)
- Improve support for UNC paths in Remote CSS. Connect now attempts to auto-correct UNC paths with the incorrect number of slashes. (SHARED-65308)
- **Zoom** options improved. (SHARED-64880/61294/61296)
- Apply button added to the Stylesheet **New Rule** dialog. (SHARED-60852)
- Fixed an issue whereby formatting could be lost after invoking undo or redo when the source tab is active. (SHARED-64734)

Connect 2018.2 Output updates

Important Information

The Output Engine (Weaver) memory footprint has increased in Connect 2018.2. This means some users might need to increase their Weaver memory allocation in the **Engine Setup** preferences.

Note: When running really large jobs, it often pays to increase Weaver memory allocation, even if only for the duration of the production job(s).

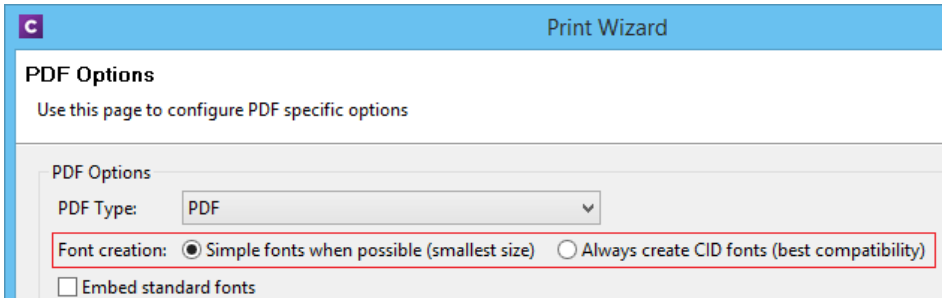
Font handling

- **Additional Content Text improvements**
  When adding text as Additional Content in Output Presets, TrueType Collections (TTC) font files were not supported. TTC fonts allow a single file to hold multiple fonts types. Since TTC fonts can be important for Asian language fonts, we have now added support for these.

  Previously, Additional Content text allowed selecting Bold and Italic for any font, even if the bold or italic version was not present. We now only display the font variation options for a font, if that font variation is actually available.
- **PDF Output font selection**
  Added selection for either simple mixed byte font encoding (for smaller size) or fixed 2 byte CID fonts with identity encoding (for greatest compatibility) for PDF output. (SHARED-65462)

  Connect always used simple fonts in its output, but dynamically switched to composite fonts when the number of characters in a font grew too large. This approach best reduced output file size. But in some cases the dynamic font creation caused viewing issues with certain PDF viewers, even though the font method is compliant with PDF standards.

To accommodate such, we have provided an alternative font creation method which always creates CID fonts with identity encoding. This results in larger output files, but provides better compatibility with PDF viewers.

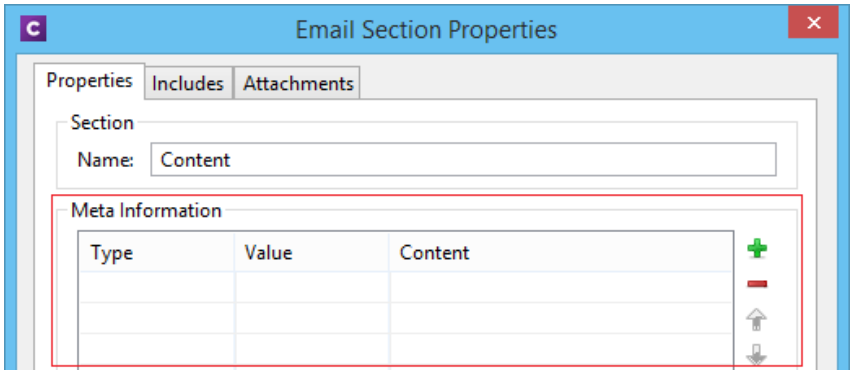


When generating PDF for non-Western languages, this new option may be the recommended choice.

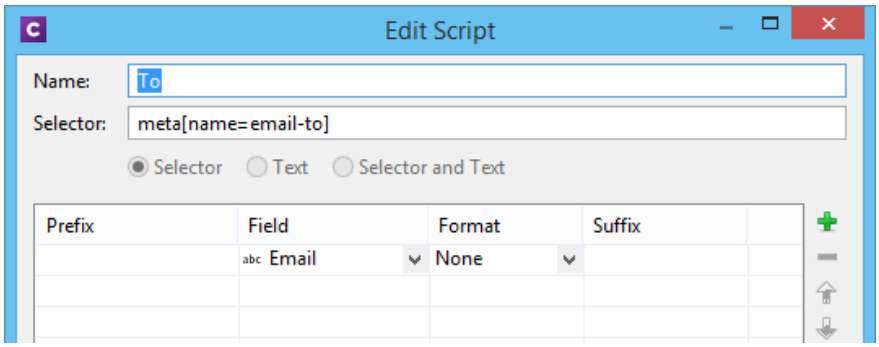
- Asian language PDF output made smaller through improved glyph handling. (SHARED-63549)
- Further improvements made in handling simulated style fonts for OTF fonts. (SHARED-65402)
- Fixed an issue with output of TrueType font encoding information that could cause special characters to be missing or incorrectly substituted. (SHARED-62332)
- When a TrueType font "glyph count" information is incorrect Connect will now attempt to continue processing, rather than immediately throwing an exception. (SHARED-64382)

Email Output Improvements

- **Meta information added to the Email Section Properties**
  This allows viewpoint meta information to be added and configured in the email. This meta information will not be visible to the receiver, but can have an effect on how the email is represented in the email client. (SHARED-64761)



- **Email Script dialog improvements**
  The email Script dialogs have been streamlined and simplified.



  Improvements were made to the **To**, **CC**, **BCC**, **Reply-To**, **From**, **Subject** and **PDF Password** email scripts. (SHARED-61005)

- **Improved support for graphs in emails**
  As email clients don't usually support SVG images, graphs added to an email context are now rasterized by default. There is also a preflight message for emails containing graphs set without rasterize options. (SHARED-60546)

Print Output Improvements

- **AFP output speed improvement** for image intensive jobs containing lots of pages. (SHARED-55831)
- If a template containing **linked annotations** (either URL links or email links) was printed to PDF as PDF/A-1b, PDF/X4 or PDF/VT, the resulting output file was not compliant to the specified conformance level. This has been fixed. (SHARED-61714)
- We introduced a way to create **custom TLE** and **NOP records** in AFP output, through a custom printer definition. These records can have variable content, and they can be created conditionally. At the moment there is no GUI option to create these custom records and their rules, but this is intended for a future version. (SHARED-65527)
- In PDF/A-1b, PDF/X4 and PDF/VT output the document title in the PDF metadata is now derived from the template name, rather than the full path of the output file name. (SHARED-61168)

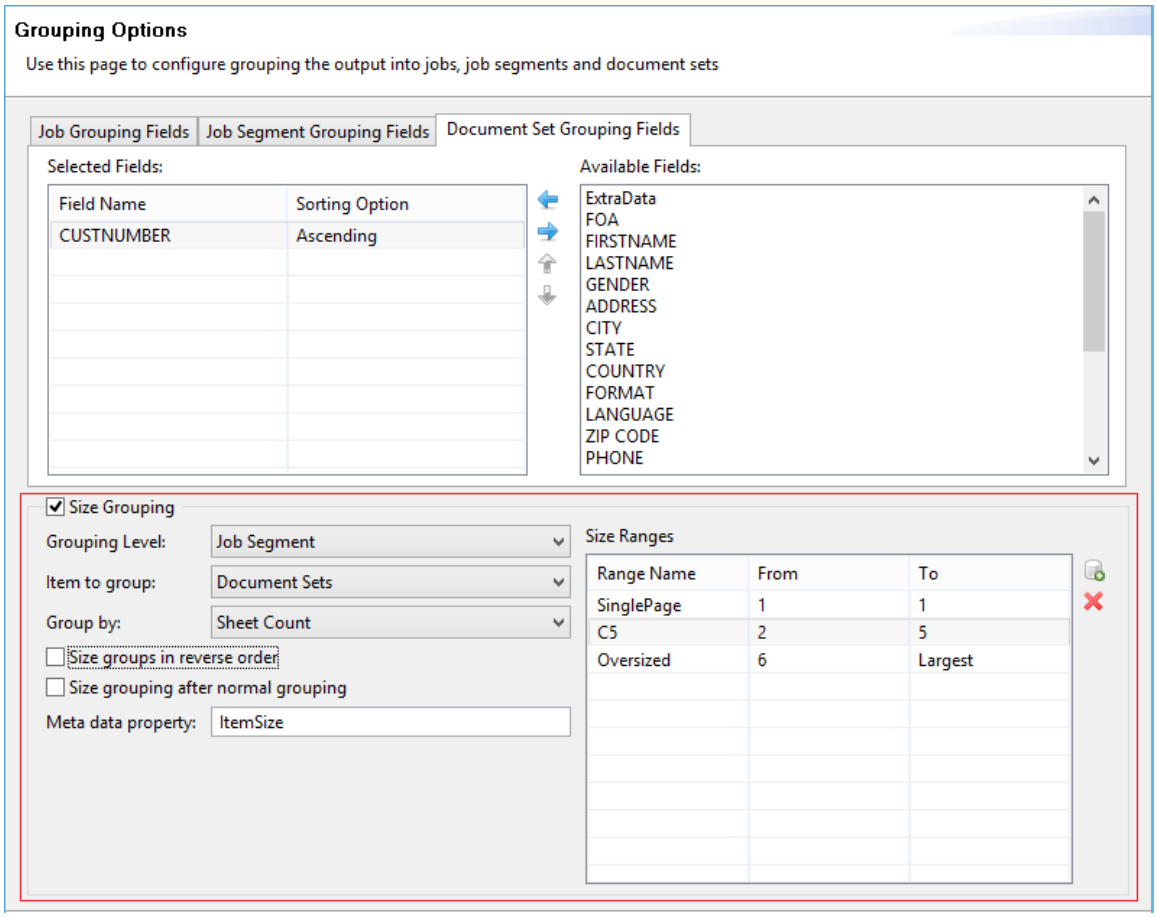
General Output Improvements

- Output engine now warns instead of crashing if it encounters text with a non-invertible transformation matrix (such as a zero point size). (SHARED-59744)
- Output engine now logs a warning if an image in an encrypted PDF file cannot pass-through untouched. (SHARED-59802)
- Fixed uncontrolled appearance of optional content (such as virtual stationery) in documents that are processed by the transparency flattener. (SHARED-60923)
- Fixed handling of output paths with one or more dollar signs ($) in them, such as certain network shares. (SHARED-63872)
- Several miscellaneous improvements to memory management.

Print Wizard and Preset Wizard Improvements

Size grouping

The ability to group documents by their size has been extended. In addition to grouping documents, you can now also group *document sets* and *job segments* by the number of pages. And instead of just using the page count, it is also now possible to use the sheet count for the size of the documents, sets, or segments. This makes size grouping useful for jobs that might contain both simplex and duplex content. (SHARED-61192)
This was previously referred to as “page break grouping”, and is found on the same Grouping Options Wizard page.



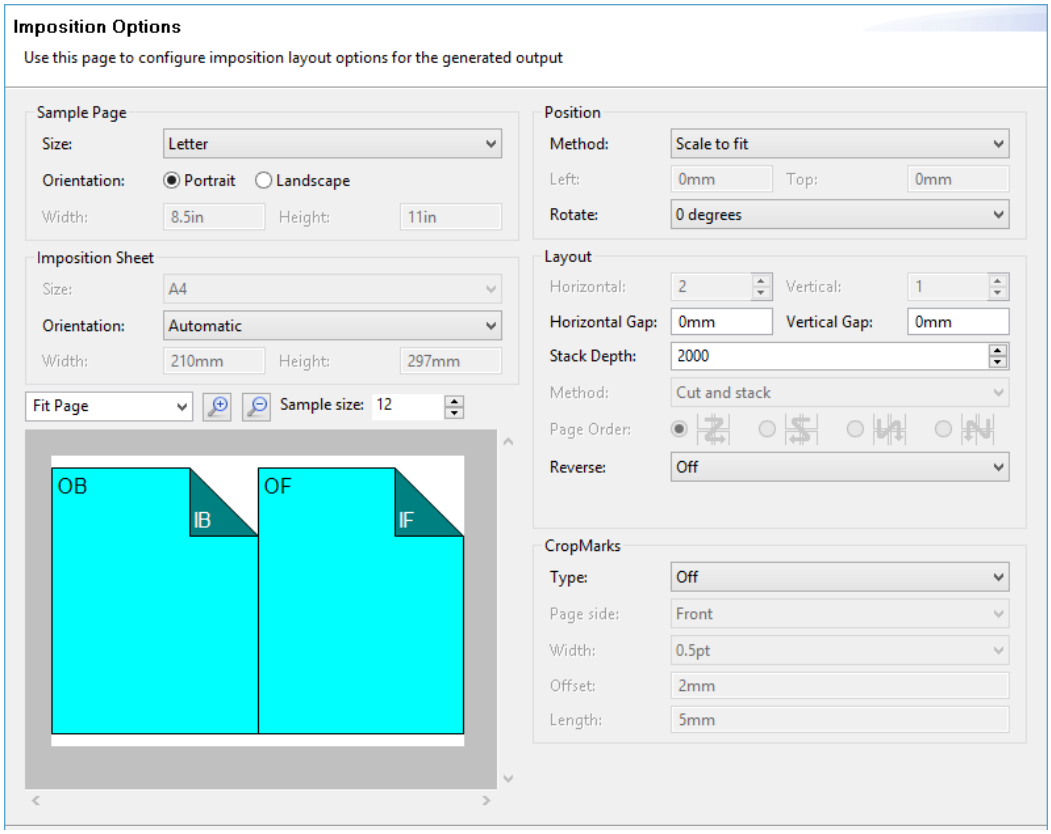
The settings used in the screenshot above groups documents into document sets by their customer number. Size grouping then creates three different jobs segments: the first with sets of only 1 sheet, the second with sets of 2 to 5 sheets, and a third with anything of 6 sheets or greater.

Tip

In Output Creation, you could create an output file per job segment, and use `${segment.metadata.ItemSize}` in the output file mask to automatically name these files appropriately.

Impositioning

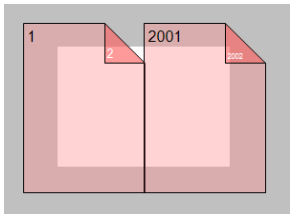
We have made a number of improvements to Impositioning, both by improving usability and by adding new features. These changes led to a redesign of the layout of the imposition wizard page, with some options being moved and others renamed.



The major improvements are as follows.

Visual warning

The page layout diagram now displays in red when the selected imposition options do not fit the selected sheet.



Sample page for imposition

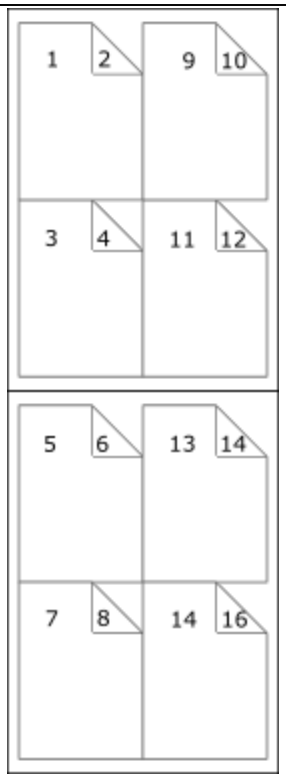
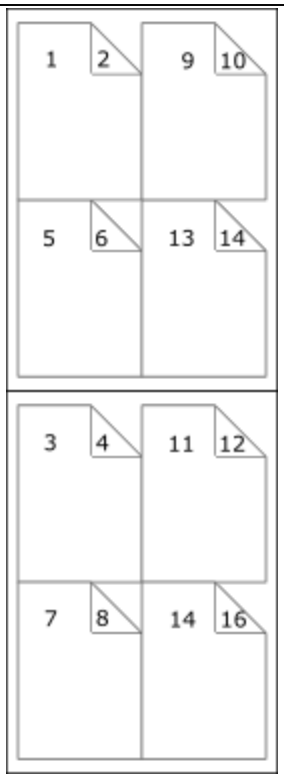
The imposition settings are not always meant to work with just your current template. So we

have added a sample page setting that will be used for both the preview and the validation of the settings. This sample page now defaults to the size and orientation of the first section of a template, instead of its first media.

Stack by Column

Alongside the existing cut and stack impositioning order, we have introduced an imposition order that is more suited for continuous feed printing when the imposition has multiple rows. We call this new ordering “Stack by Column”, and it is very suitable for roll fed label printing.
It works like this: instead of always positioning the subsequent page on the next sheet (until the stack depth is reached), stack by column first goes to the next row down on the sheet, and then down in the stack. The effect is that if all sheets are layed out top to bottom consecutively, the impositioned pages are ordered by column. If the imposition has only 1 row, both methods are identical.

Comparison of Stack by Column and Cut and Stack with 16 duplex pages in a 2×2 imposition.

Stack by Column (top to bottom, stack down, left to right)	**Cut and Stack** (stack down, top to bottom, left to right)
	

Reverse impositioning

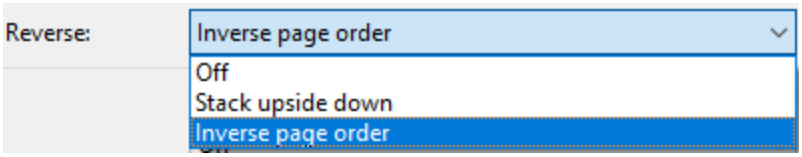
In roll fed applications, reversing the imposition order can be very important, because, when

done properly, this causes the roll output by the printer to have the first page on the outside of the roll, allowing the roll to start with the first page.

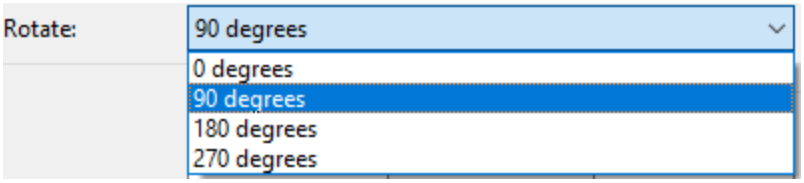
Our existing option to reverse the order of impositioning is reversing the order of the resulting imposition sheets by starting with the last side of the last sheet, and then going backwards. This effectively turns the stack up side down. It also means that, in case of duplex impositioning, all back sides become front sides and vice versa.

To have an effective way of reversing a Stack by Column imposition that leaves the “by column” order in tact, but that starts at the last pages in the job, we have added a new option to reverse the imposition, that basically reverses the incoming sheet order before impositioning. The effect is that not only is the order of the imposition sheets reversed, but the order of the pages on the sheet is also reversed.

To properly distinguish between these two ways of reversing, we have renamed our existing reverse to “**Stack upside down**”, and our new way of reversing is called “**Inverse page order**”.



Rotate imposition
In addition to our existing options of having your imposition either upright or rotated 180 degrees upside down, we can now also rotate your imposition by 90 or 270 degrees.



Selective Inserts

Selective inserts in the *Inserter Mark Options* page now support metadata menus for inserting metadata field names in selective insert conditions. (SHARED-61152)

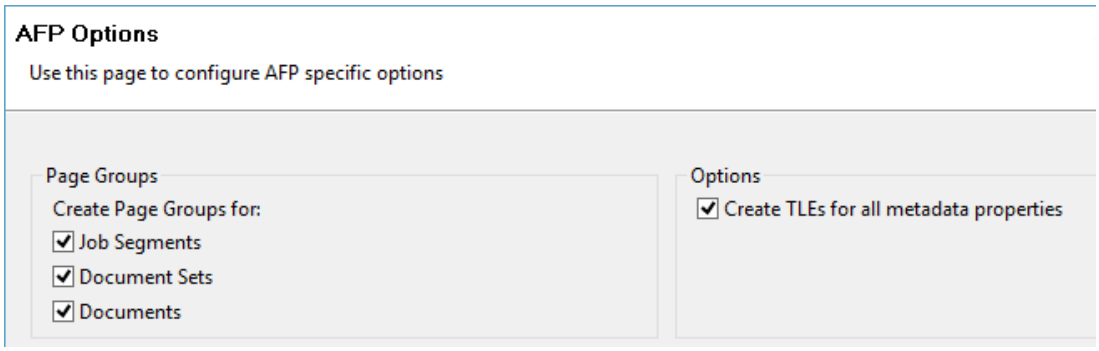
New AFP Options

We have introduced a new **AFP Options** page, to provide greater control over AFP output. (SHARED-63234)

Connect already had the ability to add metadata properties from Job Creation as TLEs to the AFP output. However, the earlier implementation did not support creating TLEs for specific pages, or for adding TLEs at the page level (which allows different values for each page).

TLEs are often used to mark the start of documents or mail pieces, to indicate media types, locations for inserts (including the name of the insert), and other properties required for print production. To accommodate this, it is now possible to conditionally create both TLEs and NOPs in AFP output by using inserts in custom printer definitions.

In addition to this, we have also added settings to control which job elements page groups will be created in the output. Earlier Connect versions would always create page groups for all job segments, document sets and documents, which would lead to pages being nested in three levels of page groups. Now you can choose to have just one level of page groups, or even no page groups at all.

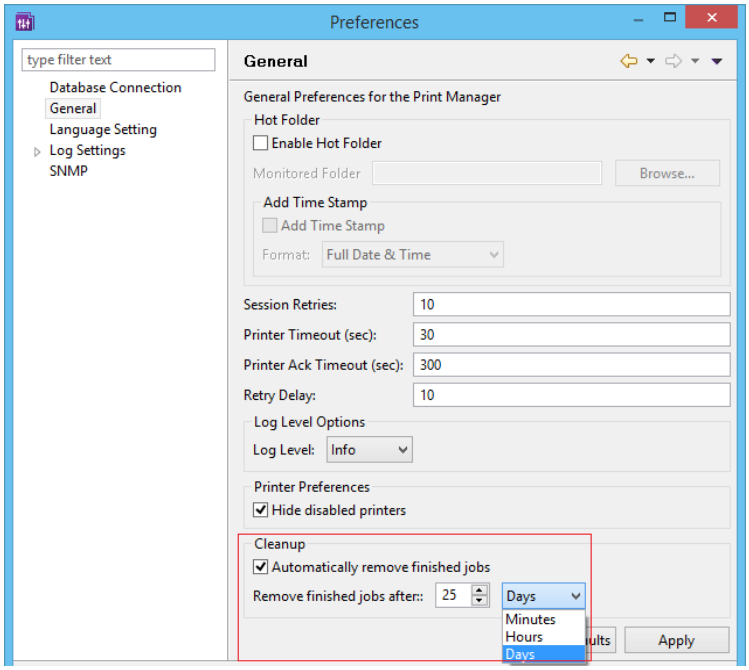


To make sure that metadata properties from Job Creation do not interfere with dynamic TLEs from the printer definition inserts, this page also has a setting to turn off TLE creation for metadata properties.

Connect 2018.2 Print Manager updates

Job Cleanup Improvements

Option added to have finished Print Manager jobs automatically deleted after a set period of time. (SHARED-62981)



General Print Manager Improvements

- **Date formats** made consistent in the *History* tab, to assist in date searching. (SHARED-59679)
- **Text files** (*.txt) are now treated the same as comma separated text files (*.csv) in Print Manager. (SHARED-64666)
- Fixed issue with **IPDS Preview** of GOCA content sometimes displaying random lines across the page. (SHARED-63331)
- Fixed issue with printer connection not always closing after querying printer capabilities. This sometimes led to failures in subsequent printer capability queries. (SHARED-64498)

Known Issues

As of PrintShop Mail Connect 2018.1 the Known Issues have moved from the Release Notes to the Connect online Help.

They can be found here.

PrintShop Mail Connect Release Notes 2018.1.6

Overview

This document provides an overview of the new features and enhancements in PrintShop Mail Connect 2018.1, as well as some important installation information.

Installing PrintShop Mail Connect 2018.1

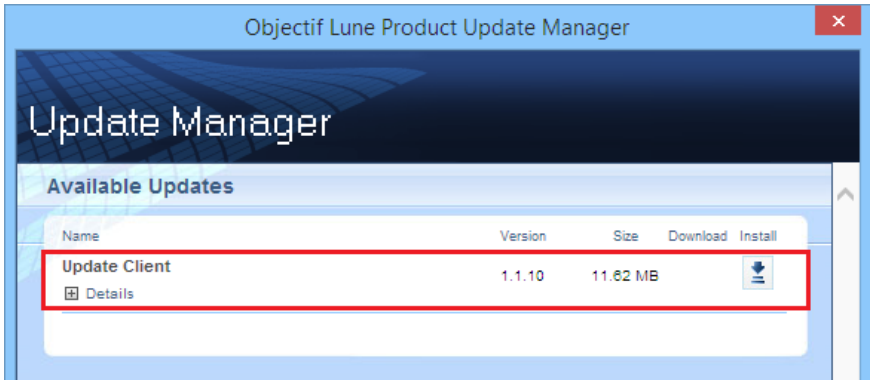
- PrintShop Mail Connect is released as a 64 Bit version only.
- Full details on installing and licensing PrintShop Mail Connect can be found in the online help, which can be accessed from the software and the installer.
- Note that PrintShop Mail Connect comes with a 30 day trial license by default.

Upgrading from PrintShop Mail Connect 1.7

It is highly recommended that you update the **Objectif Lune Update Client** before upgrading PrintShop Mail Connect from version 1.7 to version 2018.1.

If you do not update the Update Client, an unexpected error might occur whilst updating Connect. This error does ***not*** prevent the successful upgrade of Connect to 2018.1, even though it appears as if it might have. To avoid potential confusion, we recommend that you first update the **Objectif Lune Update Client** before attempting to upgrade Connect from version 1.7 to version 2018.1.

The Update Client will show that there is an update available for itself. Simply click on the download button in the dialog to install the new version of the Update Client. Note that it is no problem in running the update while the Client itself is still open. It will automatically update itself.



Connect 2018.1.6 Enhancements/Fixes

Improved Weaver Engine memory usage

Weaver Engine memory usage has been considerably improved. These changes have resulted in the Weaver Engines not only requiring fewer system resources but have also improved stability and reduced the need for periodic restarts. (SHARED-65330)

Improved font handling

- Improved font style simulation, for when the full font family is not available.
  Note that this might cause some very minor output differences in comparison to earlier Connect 2018.1 font style simulations. (SHARED-65336)
- Incompatibilities between the Windows API and some custom fonts that required font style simulation could lead to Merge Engine failures. These issues have now been addressed. (SHARED-65334)

Blank pages encountered in jobs with PDF backgrounds of 30 pages or more

An issue was discovered in jobs in which the PDF background input exceeded 30 pages. In this scenario some pages would output blank after the 30 page limit was reached. This error was due to the internal image cache being exhausted and has now been fixed. (SHARED-65209)

Dynamic Printer Control not applying some rules correctly

Dynamic PPD rules that were supposed to apply to Job/File level were actually being applied at the Page/Sheet level. This error has now been fixed.
(SHARED-65328)

Connect 2018.1.5 Enhancements/Fixes

Blank page inserted in some multi-page transactional tables

(PReS Connect and PlanetPress Connect only)

Issues were encountered with some transactional print jobs, whereby a blank page would be inserted towards the end of some multi-page transactional records. This problem would only be encountered under very specific circumstances. Such as in duplex transactional documents with records that exceeded 15 pages, or where two documents with page numbers in the mid teens followed each other. These issues have now been fixed. (SHARED-64556)

Connect 2018.1.4 Enhancements/Fixes

PDF/A-1b compliance broken with Polish text

An issue was discovered with Polish text in PDF/A-1b compliant output, whereby the output was missing "*CIDSet*" font descriptor entries for TrueType-based CID fonts.
This error has now been fixed. (SHARED-64536/64816)

Issues with text extraction from PDF output

Problems were encountered in extracting non-latin and multilingual text from PDF output. These problems have now been addressed. (SHARED-64936)

Connect 2018.1.3 Enhancements/Fixes

Hyperlinks broken in PDF output

Hyperlinks in PDF output were broken in 2018.1. This issue has been fixed, and hyperlinks now work as expected in 2018.1.3. (SHARED-64353)

Connect 2018.1.2 Enhancements/Fixes

Issues running Connect jobs through Enhance

- When running multiple parallel output channels via Enhance configurations it was possible that some output channels could attempt to write the exact same content at the same time, thus using the same resources. This could lead to conflicts and resulted in errors and inconsistencies between outputs. This issue has now been fixed. (SHARED-63372)

- When creating PostScript output using the REST API to execute an Enhance configuration, the resulting PostScript file contained incorrect DSC comments if the name of the media used on a page contained one or more space characters. This has been fixed. (SHARED-64334)
- The PostNL KIX database has been updated to the latest 2018 version.
  Note: The KIX database is only available when using the REST API to execute Enhance configurations using the KIX functionality. Standard Connect usage does not allow use of the KIX database. (SHARED-64358)

OutOfMemory Errors in PostScript Output

When creating PostScript output or using the REST API to run a custom Enhance configuration containing one or more PostScript output channels, "*OutOfMemoryError: Metaspace*" errors could be encountered. This was due to JavaScript source code in the PostScript printer definition being compiled too often. This issue has now been fixed. (SHARED-64350)

Issue with background image URL paths

Fixed issues with URL path encoding. This issue applied to Section background image paths (\\host\path\image.png) containing special characters, such as spaces and hash signs ('#'). (SHARED-64528)

Special characters displayed incorrectly in Designer Snippet editor

Fixed an issue with the encoding of non-Latin-1 characters in HTML snippet editors. (SHARED-64530)

Connect 2018.1.1 Enhancements/Fixes

Section cloning issues

Two separate Section cloning issues were fixed. (SHARED-64087)
The issues were:

- Cloned content was being based upon the section for which *addBefore* or *addAfter* was invoked, rather than the section the clone was originally based on.
- The rendered page count could be incorrect. This was due to the page count of a section bundle (consisting of sections with continuous page numbering) mistakenly overwriting the page count of the previous section bundle.

Weaver Engine memory leak

Improvements made to how the Weaver Engine reclaims memory, in order to reduce any memory leakage. (SHARED-64205)

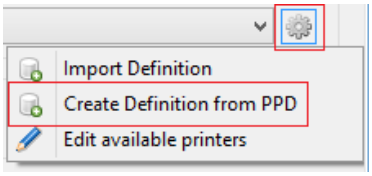
Issue with Type 3 fonts in PDF Output

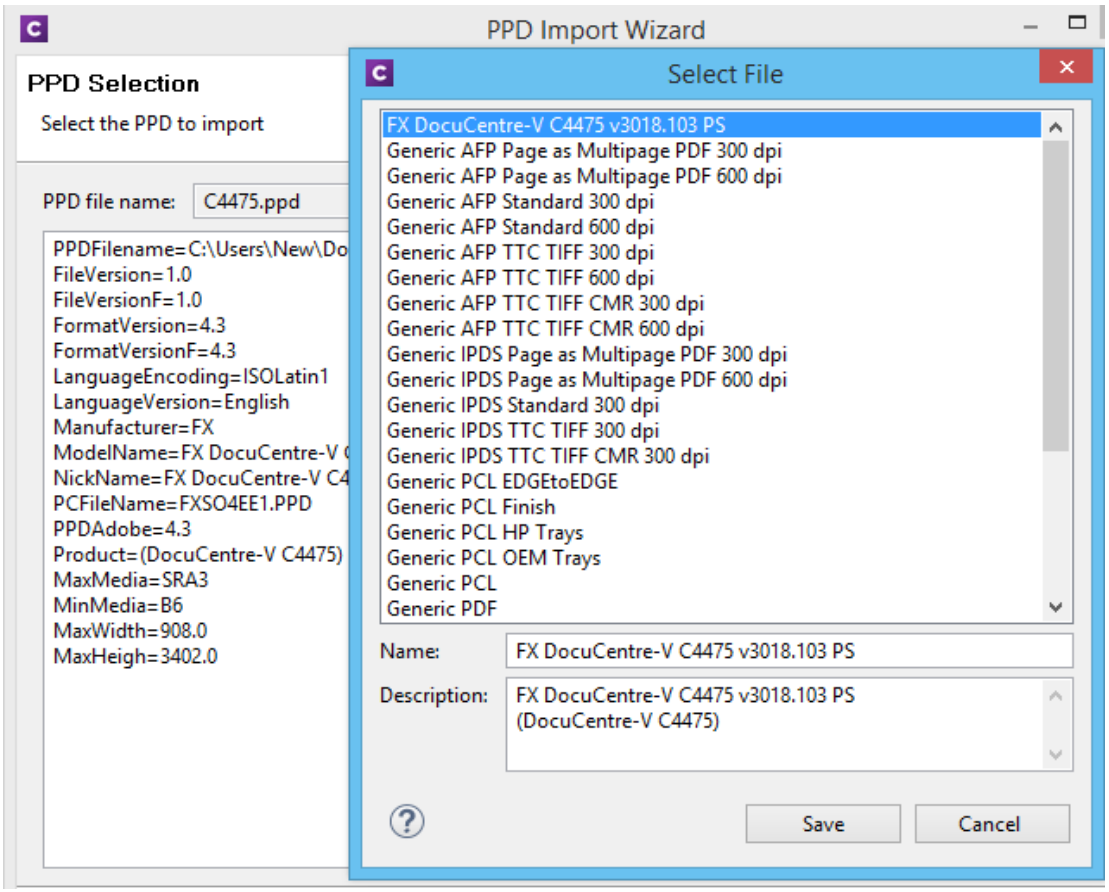
Type 3 fonts not embedded in PDF Output are replaced with font substitutions. These could become unmanageable in some circumstances. This issue has now been fixed. (SHARED-64214)

Connect 2018.1 General Enhancements

Dynamic Print Control for PostScript printers

Connect 2018.1 introduces a new way to drive PostScript printers. Individual PostScript printers can now be selected in the Printer Wizard and Output Preset via their associated PostScript Printer Definition (PPD) files:





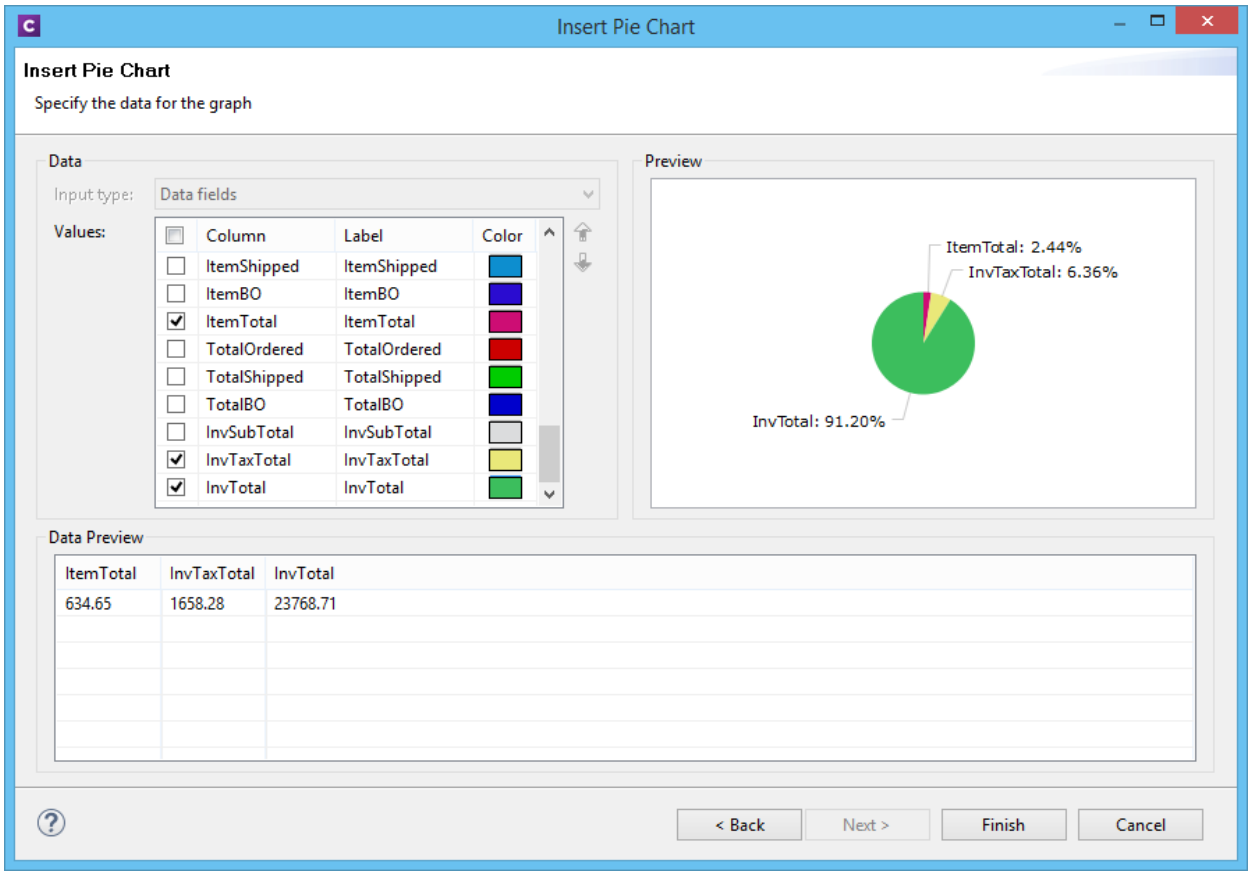
This allows:

- The use of printer features that are not yet supported through the Connect user interface, such as:
  - switching between color and greyscale printing
  - selecting output bins
- Direct control of supported PostScript functionality, such as:
  - Duplex/Simplex
  - Print Media selection
  - Finishing options (e.g. Stapling, Binding and Folding)

NOTE: Case sensitivity/insensitivity has not been added as an option in this first cut of the Dynamic Print Control rules editor. This will be added in a subsequent version.

Improved Business Graphics

Business Graphics have been greatly enhanced and now display a preview of the data and graph as part of the Pie, Line and Bar chart wizards. The updated wizards simplify data selection and allows you to set and preview graph options dynamically, prior to inserting the graph object. This same functionality has been added to the scripting wizard, which can be launched from any Business Graphic script in the Scripts panel:



The Properties dialog in Connect lets you configure the graph via the User Interface and stores the configuration according to the specifications for this library. A Source tab has been added to the Graph Properties dialog so that you can view and edit the JSON configuration.

The implementation now supports the full Line, Bar and Pie chart API of the underlaying amChart library. This allows you to enable amCharts graph features that are not available in the user interface of Connect.

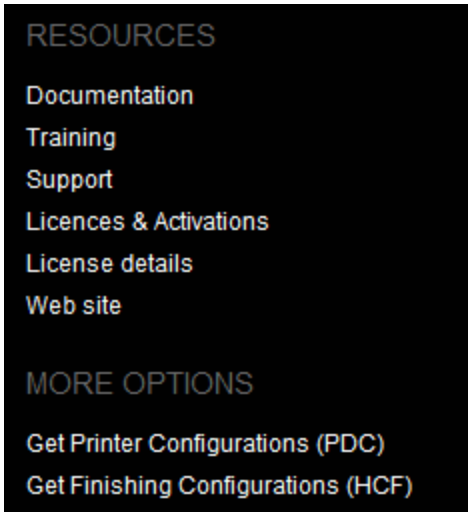
It is also possible to copy/paste Line, Bar and Pie chart configuration settings from the online amChart Live Editor directly to this tab.

Refer to the amChart documentation for more information on the numerous configuration options available.

NOTE: As a consequence of changes in both the user interface and the underlying technology, Business Graphics made with a version prior to PrintShop Mail Connect 2018.1 may not display correctly when opened in version 2018.1. See the ***Business Graphics: Backwards Compatibility*** section on this page for full details.

Welcome Screen improvements

The Welcome screen layout has been redesigned and streamlined to simplify navigation and improve access to Connect resources, including current license details. (SHARED-61377)



Connect version numbering changed

We have changed the version numbering system. From here on in, the version will be year based, with the second number showing the major release and subsequent numbers showing minor releases. The intention is to have two major releases a year, with minor releases in between to address specific customer issues.

2018.1 is the first of the two planned major releases for 2018. The second will have the version number 2018.2. Any minor releases between the two major releases will be labelled 2018.1.n, where "n" is a numeric counter. Thus the first minor release would be 2018.1.1 and the second would be 2018.1.2, and so forth. (SHARED-60069)

Windows Server 2016 now officially supported

As of PrintShop Mail Connect 2018.1, Connect is now officially supported under Windows Server 2016.

NOTE: The Upland Objectif Lune **Update Client** application might be blocked by enhanced security settings in Windows Server 2016. To fix this, add http://updates.ca.objectiflune.com to the list of trusted web sites on that machine, or lower the internet access rules.

Improved Previews

The Preview now smooths text, images and line art when rasterizing a page. This improves the quality of the resulting image, especially when rasterizing a page with lower resolutions. (SHARED-59724)

Localization improvements

Improved localization and translations throughout PrintShop Mail Connect.

Improved Data handling:

- Improvements to the internal handling of data field names to support data fields that do not conform to a standard SQL query syntax. This means that fields with characters such as #, $, % and ^ can now be processed. (SHARED-59126)
- Performance improvements made to the Connect MySQL back-end database, through changes made to internal database indexing. (SHARED-59393)

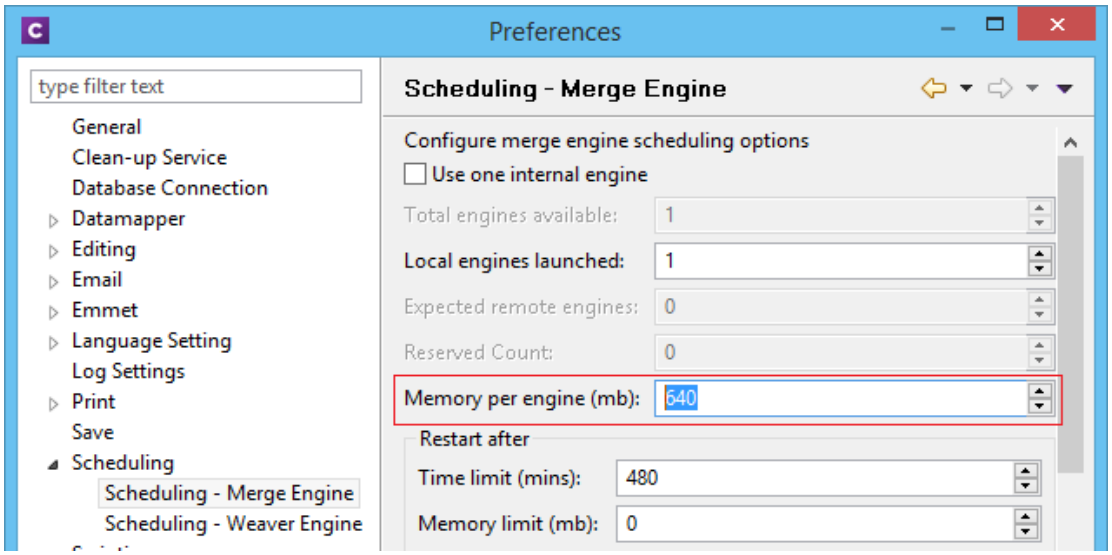
Improved support for Microsoft SQL Server back-end

A number of enhancements have been made to improve the performance and support for Microsoft SQL Server (MS-SQL) as the Connect back-end database. These include:

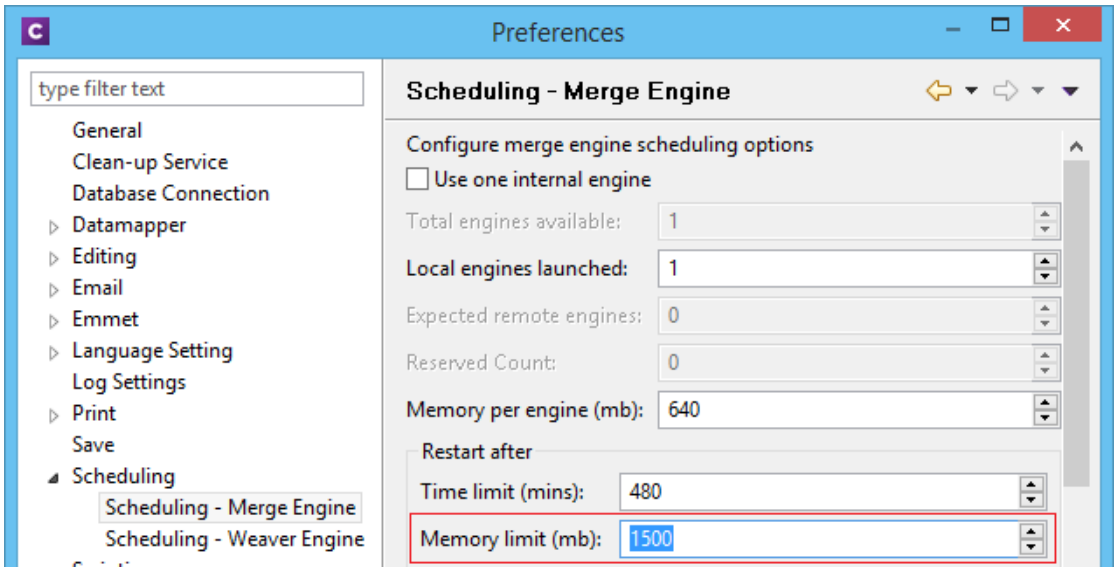
- Database Table Partitioning, significantly speeding up the clean-up service. (SHARED-59589)
- Internal Connect SQL statements re-factored to work more efficiently with MS-SQL. Improving support for large jobs, in particular. (SHARED-57575)
- MS-SQL driver updated. (SHARED-60464)

New Preferences options:

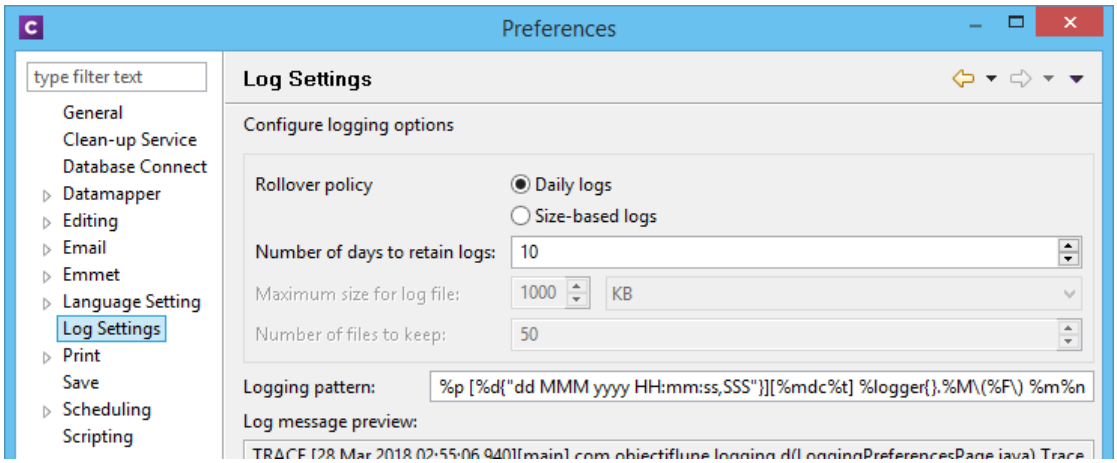
- Scheduling preferences for Merge and Weaver engines now include a preference for specifying the memory (RAM) to be used per engine. (SHARED-44547)



- Merge and Weaver engine scheduling preferences now include an option for scheduling an engine restart if the total amount of memory (RAM) used by the engine exceeds the specified entry. This is in addition to the time limit setting. (SHARED-63101)



- Log file settings can now be adjusted within the Designer, Print Manager and Server Configuration preference windows. The settings are global to all Connect applications. Settings include log file rollover options (time or size based), and the format of the log messages generated. (SHARED-60043)



Installer Improvements:

- Prerequisites now installed silently when installing Connect using silent installer. (SHARED-59286)
- Messaging improved for when the Connect installer cannot safely commence un-installation. (SHARED-59565)
- Installation folders cleaned up. Those that were no longer necessary have been removed, and others have been renamed or moved to make them more logically consistent.

Back-end Improvements:

- Better handling of non-responsive engines. (SHARED-55305)
- Improved response to Merge engine failures. (SHARED-46180)
- Reduced memory footprint when running the same job multiple times. (SHARED-57905)

Logging Improvements:

- Improvements made to Weaver logging for small output jobs and improved throughput. (SHARED-58570)

- Connect version number added to certain log files, to streamline issue identification and confirmation. (SHARED-60256)
- When using the REST API to execute a +PReS Enhance configuration that logged to a custom log file, the log file could not be deleted at the end of the execution. Repeating the execution of the +PReS Enhance configuration also caused unreadable NUL characters to appear in the log file and log messages were appended to it. This has been fixed and the custom log file now closes at the completion of the Enhance configuration execution. (SHARED-59656)
- Logging infrastructure has been updated. Logging can now be configured via an XML file located in the user's home directory under *Connect\workspace\configurations\logback.xml*. (SHARED-61913)

Connect 2018.1 Designer Enhancements/Fixes

Updated internal browser component

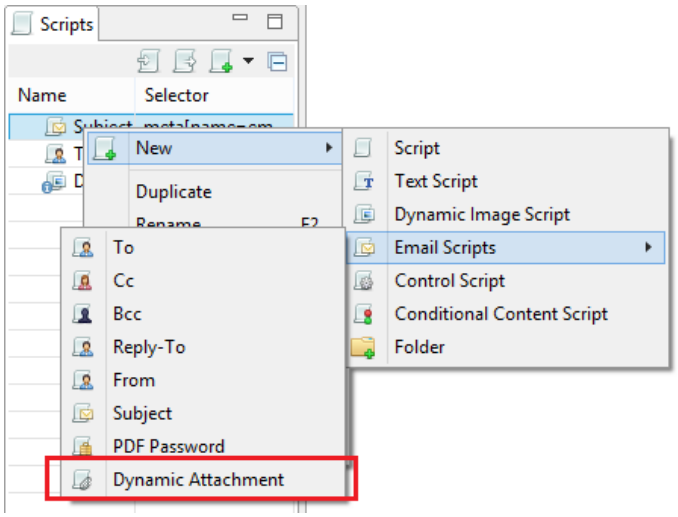
The browser component (Mozilla Gecko) used in the WYSIWYG editor of the Designer has been updated. This allows you to use new CSS properties like flexbox. Connect 2018.1 now uses Mozilla Gecko 38 as its HTML rendering engine. (SHARED-42286)

NOTE: The update to the Mozilla Gecko engine could lead to increased output file sizes for some PCL, AFPDS and IPDS jobs. This is generally not a cause for concern, however there might be some associated increase in processing times, as well as some minor differences in the output. For example, table line widths and font spacings might differ slightly (particularly for SMALL CAPS text), which could lead to slightly different word-wrapping in some cases.

Improved attachment options for Email output

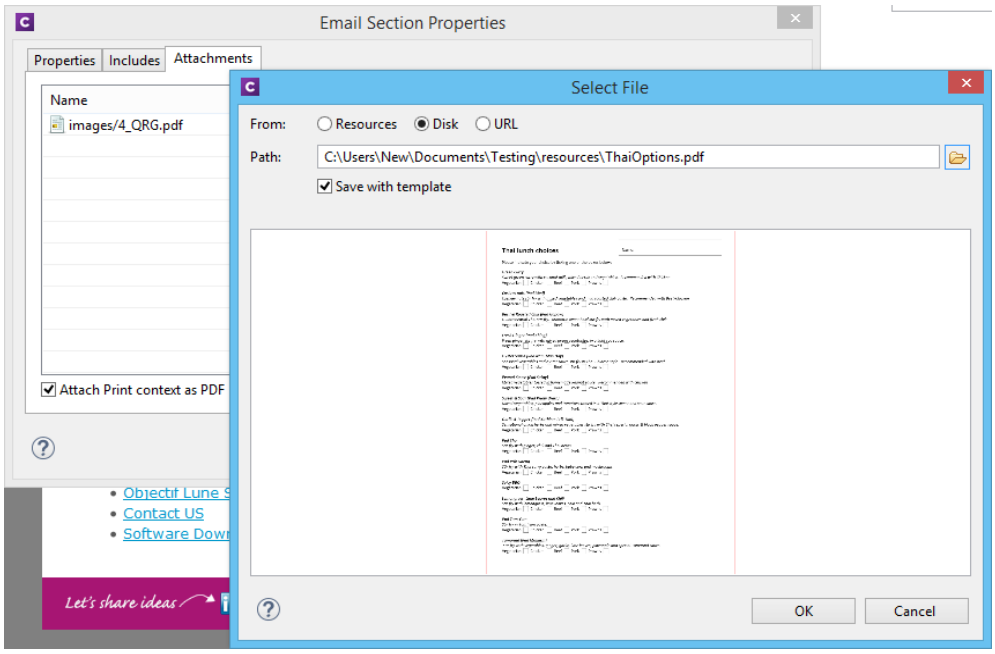
New features have been introduced to simplify the setting of email attachments within Designer.

- **Dynamic file attachments** can be added to email sections via a new *Email Scripts* option. (SHARED-56602)



  This Script wizard allows you to construct a selection using the Prefix, Field and Suffix values to dynamically specify a file from disk or a remote location.

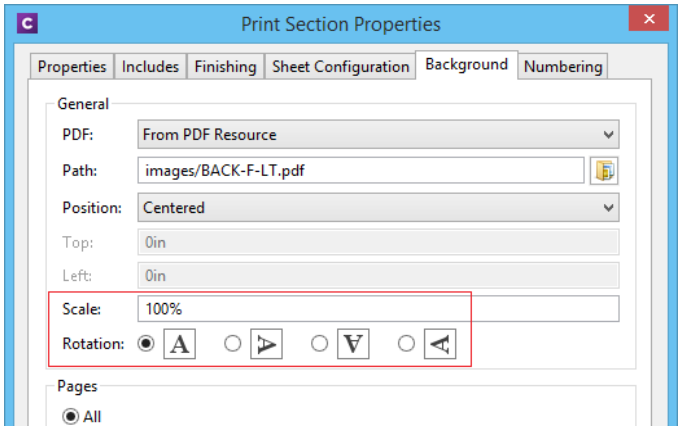
- **Static file attachments** can be added to email sections via the *Attachments* tab in the *Email Section Properties* dialog. (SHARED-28343)



You can also select whether the Print Context should be added as a PDF attachment or not, using the **"Attach Print context as PDF"** check-box.

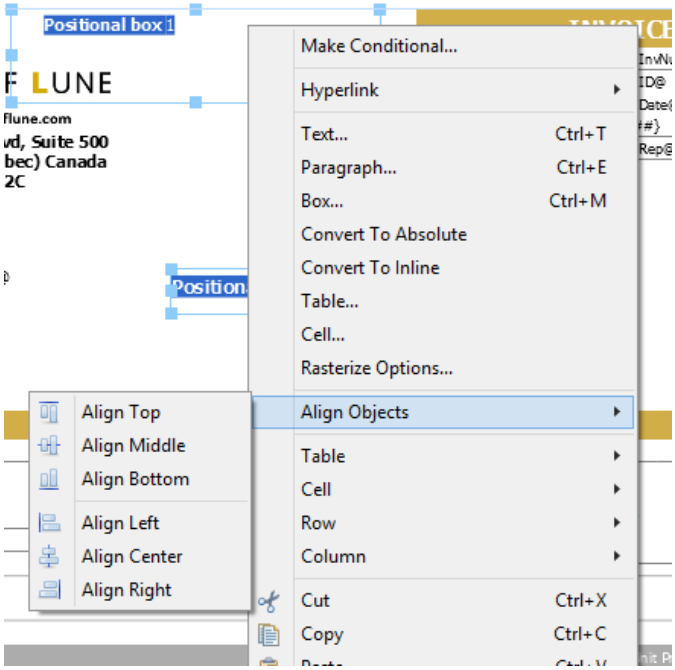
Rotate and Scale Background Images

Section Background images can now be rotated or scaled, either via the Section Properties dialog or via a control script. (SHARED-60244/60855)



Align multiple objects

You can now select and align several absolute positioned objects at once with the new **Align Objects** context menu. (SHARED-60263)



Preflight improvements

Double-clicking a script warning or error in the Preflight report will now open the offending script within a script editor. (SHARED-60273)

Improved language support

Disable certain **CTRL + ALT + <?>** shortcuts in source editors, including source tab, HTML snippet editors and CSS editors.

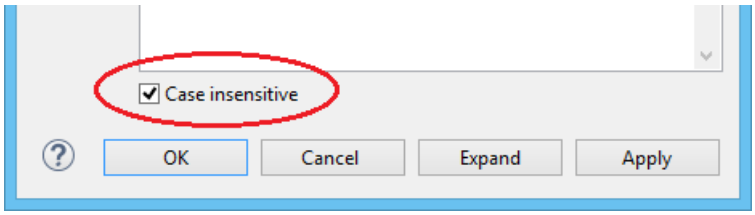
This allows adding square brackets in a source editor when using a French keyboard layout shortcut (`CTRL + ALT + [` or `CTRL + ALT + ]`), as well as semi-colons (`CTRL + ALT + ;`) on Hungarian keyboard layouts. (SHARED-60308/62056)

Data Model pane improvements

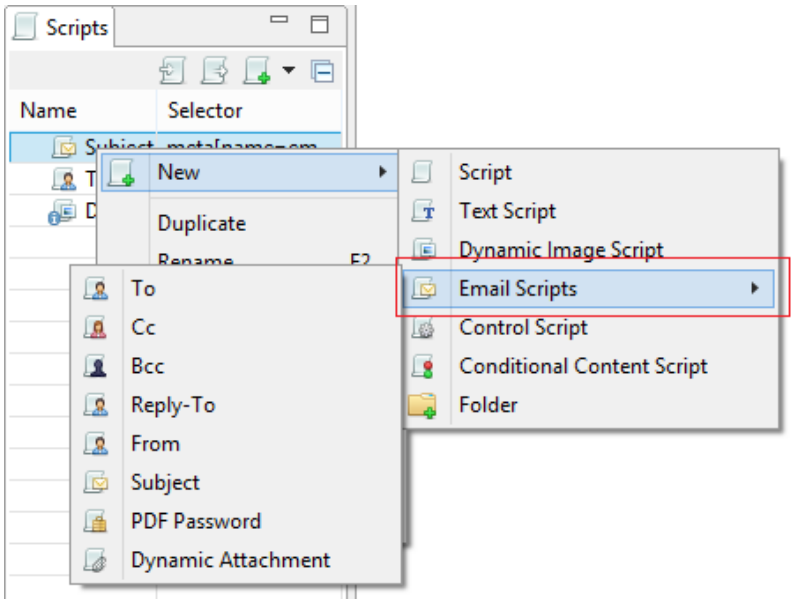
The context menu in the Data Model view now allows you to modify the data model while the template editor is active. This makes it possible for PrintShop Mail Connect users to create and modify a data model. (SHARED-60234)

Scripting improvements:

- Option added to allow Case Sensitivity to be set on or off for any conditional string comparison scripts. (SHARED-56535)



- You can now inspect Scripted elements in the Outline tree when in Preview mode. (SHARED-60231)
- New sub-folders now inherit the same execution scope as their parent. (SHARED-56964)
- You can now drag and drop a data field to a script folder even when a script of that name already exists within a different folder (or root) of the Scripts panel. (SHARED-57027)
- Improved Email Context script selection. All the Email Context scripts are now bundled together in their own sub-menu. (SHARED-56602)

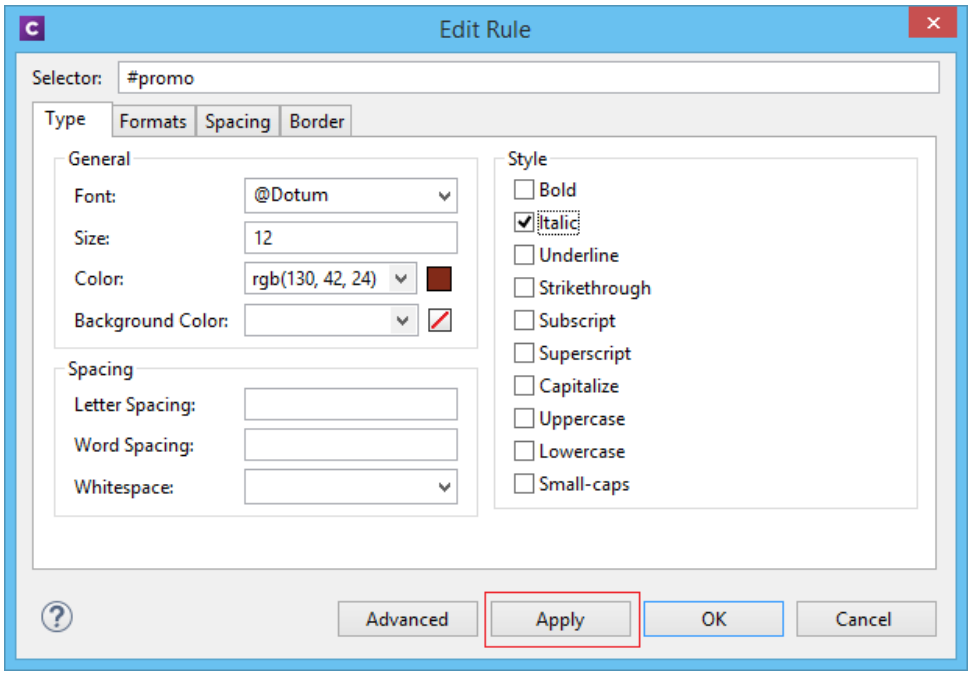


- Improvements made to the default data field selection logic when adding new Text Scripts. (SHARED-57286)
- Double clicking a script in the Script Panel with the script editor open will now prompt you to save your changes. (SHARED-47908)

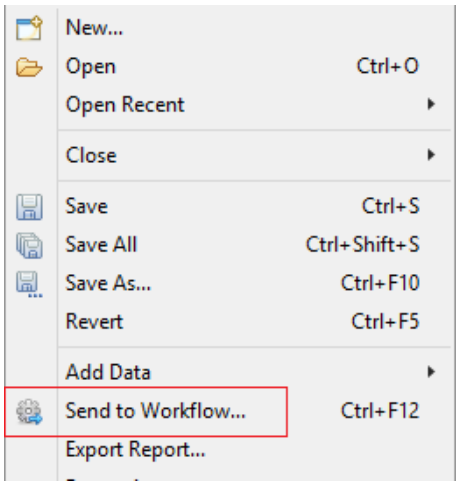
- Any open Script Editor will now close when another template is opened. (SHARED-32289)

General Designer improvements:

- Style rule changes can now be saved and applied instantly from within the Edit Rule dialog, via a new Apply button. (SHARED-58595)



- A new **Send to Workflow** icon has been added to Designer. It has been added as a new option in the icon toolbar and also to the File menu option. (SHARED-59322)



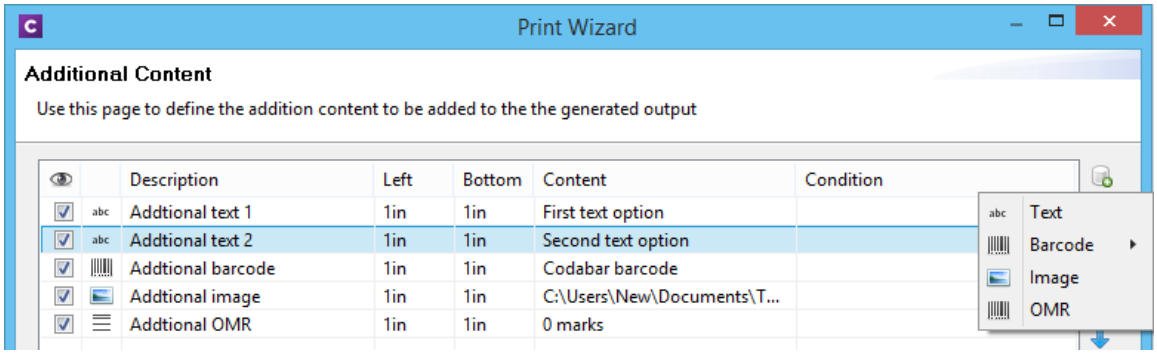
- You can now copy/paste or drag/drop images files from the file system or *Images* folder to the *Stylesheets* folder. (SHARED-49551)

- The **New Email** and **New Print** template wizards now open using the last used Template settings. Selections such as page size, page margins and the like will be retained. (SHARED-33982)
- Improved **element insertion** rules, to make context menus and dialogs more accurate. For example, the option to add Absolute elements in emails has been removed, as the email format does not support this functionality. (SHARED-60947/61117)
- Improved support for **remote images** (http-based URLs). (SHARED-58592)
- The *resource()* script function has been extended to return information about PDF permissions. (SHARED-58026)
- The attributes panel now has a "*Reset Size*" button that reverts a resized image back to its the original size. (SHARED-57025)
- You can now clear the "*recent files*" list via new **Window > Clear Recent Files Lists** menu option. (SHARED-42712)
- You can now toggle between the front and back of the selected Media entry in Master Pages. (SHARED-62038)
- You can now use cursor keys to navigate after or <article> elements that are the last element in the <body>. (SHARED-41710)
- Dialog box resizing improved and made more consistent. (SHARED-16164)

Connect 2018.1 Output Enhancements/Fixes

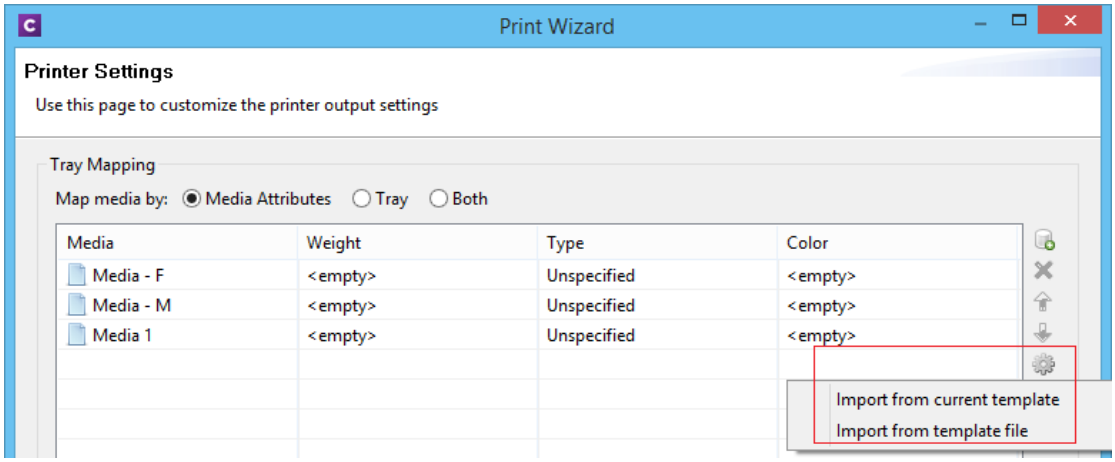
Redesigned Additional Content page in the Print Wizard

The Additional Content page in the Print Wizard has been re-designed to present all the Additional Content entries in a single table, simplifying Additional Content management. (SHARED-56444)



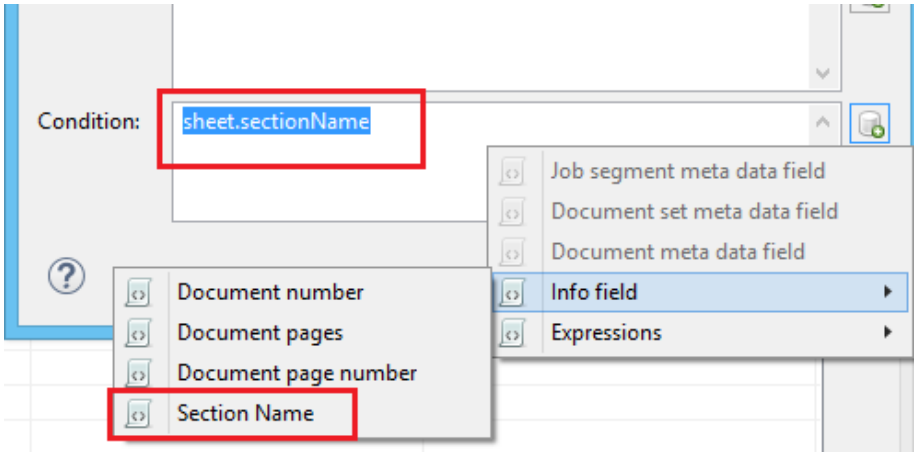
Improved Tray Mapping in Printer Wizard

Media attributes can now be imported directly into the output config tray mapping from either the currently open template or from a saved template file. (SHARED-48818)



New Conditional option

During Output Creation, it is now possible to determine the template section to which a sheet belongs to using the new `sheet.sectionName` property. This property has been made available as a conditional option in the Printer Wizard. (SHARED-61275)



NOTE: The introduction of this property introduces a slight risk of compatibility issues, but only for cases where an existing custom printer definition or Enhance configuration already has a user-defined property with the *exact same name*.

Font handling made more robust

Connect will now generate **bold**, *italic* and ***bold-italic*** variations of most fonts on machines that do not have those variants of the font installed. (SHARED-59975/60670)

Improved PDF throughput processing:

- Improvements were made to the processing and optimization of barcodes in imported PDFs. This has resulted in speed improvements when processing certain PDF files for print output. This is particularly the case for AFPDS, IPDS and PCL output. (SHARED-58782/59955)
- Enhancements made in the caching of PDF input files has led to speed improvements when processing large PDF inputs. (SHARED-58503)
- Improved memory usage with PDF pass-through. (SHARED-58782)
- PDF pass-through is now enabled by default for PDF containers in AFP and IPDS output. (SHARED-61494)

Print Output Improvements:

- Transparency support for PDF output. (SHARED-61492)
- Soft mask improvement in PDF. The soft mask is now set by the CTM ("Current Transformation Matrix") when the soft mask is created. (SHARED-58893)
- Improvements made to generic PCL output for some printers. (SHARED-60019)
- Booklet Impositioning for mix-plex jobs has been improved, reducing the likelihood of empty pages. (SHARED-60054)
- Improved Job Creation options for PrintShop Mail Connect. (SHARED-43685)

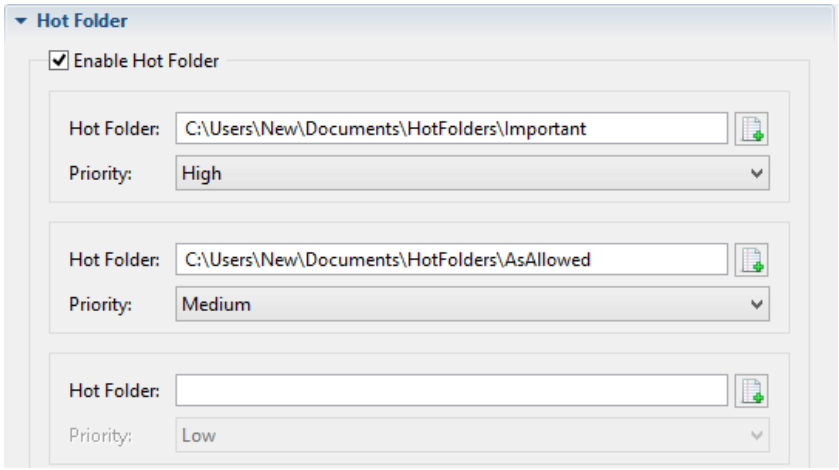
Connect 2018.1 Print Manager Enhancements/Fixes

Several improvements have been made to the Print Manager. Some to better cater for continuous IPDS printing, and others to improve useability.

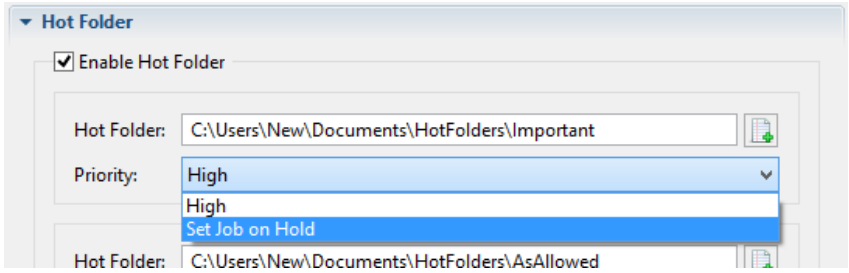
Hot Folder improvements

The Hot Folder dialog has been updated and improved. Improvements include:

- Simplified selection of Hot Folders



- Option added to put Hot Folders 'On Hold'.



  If selected, jobs will be assigned to the target printer but put 'On Hold' and not be taken from the queue until released. (SHARED-61224)
- Files are now moved from the Hot Folders to a processing sub folder and processed from there. (SHARED-60180)

General Print Manager improvements and fixes:

- Improved handling of duplicated job names. (SHARED-34710)
- Job status added to the job filename, to make it easier to identify the current job state. (SHARED-60182)
- Improvements made to file status updating. (SHARED-58122/58172)

Known Issues

As of PrintShop Mail Connect 2018.1 the Known Issues have moved from the Release Notes to the Connect online Help.

They can be found here.

PrintShop Mail Connect Release Notes 1.8

Overview

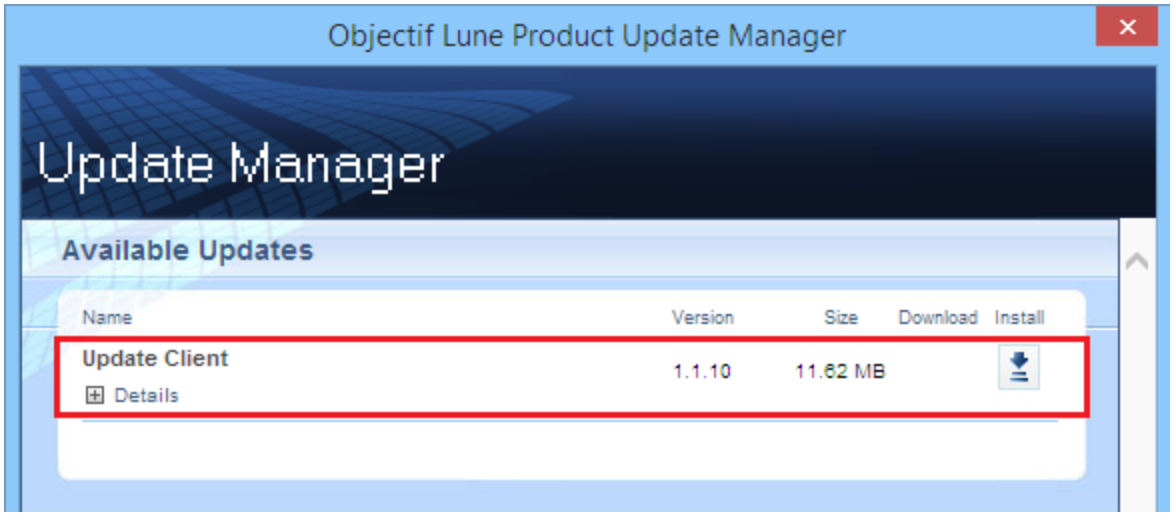
This document provides an overview of the new features and enhancements in PrintShop Mail Connect 1.8, as well as some important installation information.

Upgrading from PrintShop Mail Connect 1.7

It is highly recommended that you update the **Objectif Lune Update Client** before upgrading PrintShop Mail Connect from version 1.7 to version 1.8.

If you do not update the Update Client, an unexpected error might occur whilst updating Connect. This error does ***not*** prevent the successful upgrade of Connect to 1.8, even though it appears as if it might have. To avoid potential confusion, we recommend that you first update the **Objectif Lune Update Client** before attempting to upgrade Connect from version 1.7 to version 1.8.

The Update Client will show that there is an update available for itself. Simply click on the download button in the dialog to install the new version of the Update Client. Note that it is no problem in running the update while the Client itself is still open. It will automatically update itself.


Objectif Lune Product Update Manager
Update Manager
Available Updates
Name
Version
Size
Download
Install
Update Client
1.1.10
11.62 MB
Details

Installing PrintShop Mail Connect 1.8

- PrintShop Mail Connect is released as a 64 Bit version only.
- Full details on installing and licensing PrintShop Mail Connect can be found in the online help, which can be accessed from the software and the installer.
- Note that PrintShop Mail Connect comes with a 30 day trial license by default.

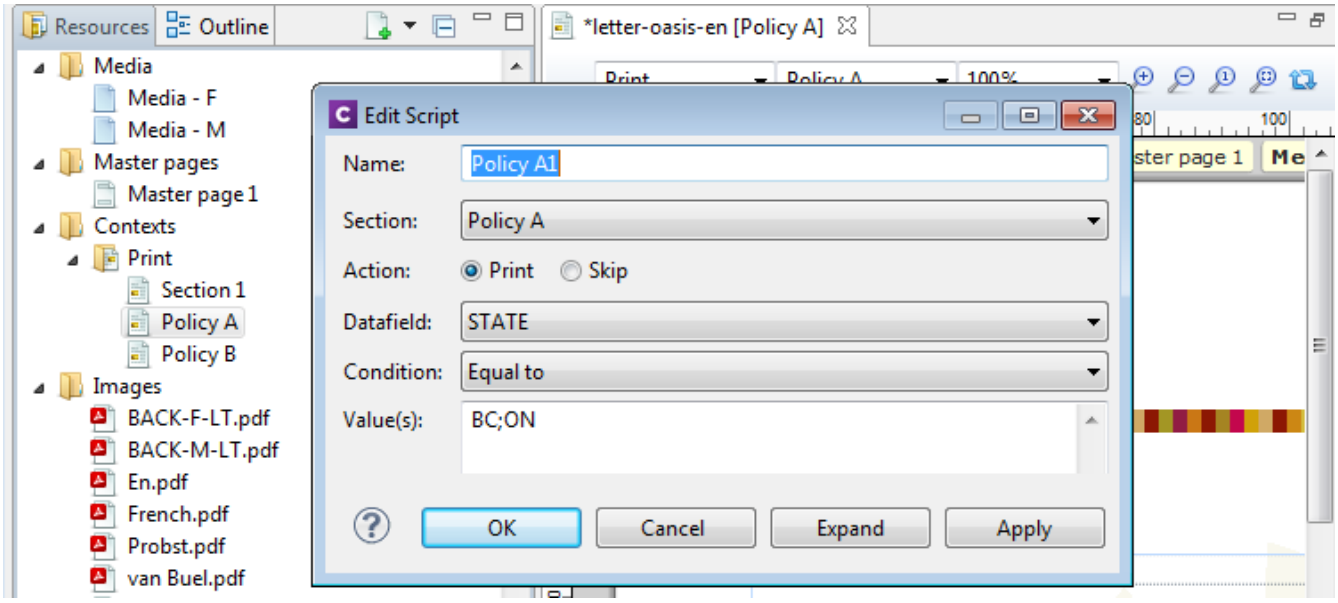
Connect 1.8 General Enhancements and Fixes

Native support for Microsoft Excel spreadsheet files

PrintShop Mail Connect can now handle Microsoft Excel files natively. The CSV data type has been enhanced to automatically recognize *.XLS and *.XLSX files and use them directly without any additional steps. Both the *CSV Wizard* and the *Add Data* options now allow you to pick these file types. (SHARED-58610)

Conditional Print Sections Wizard

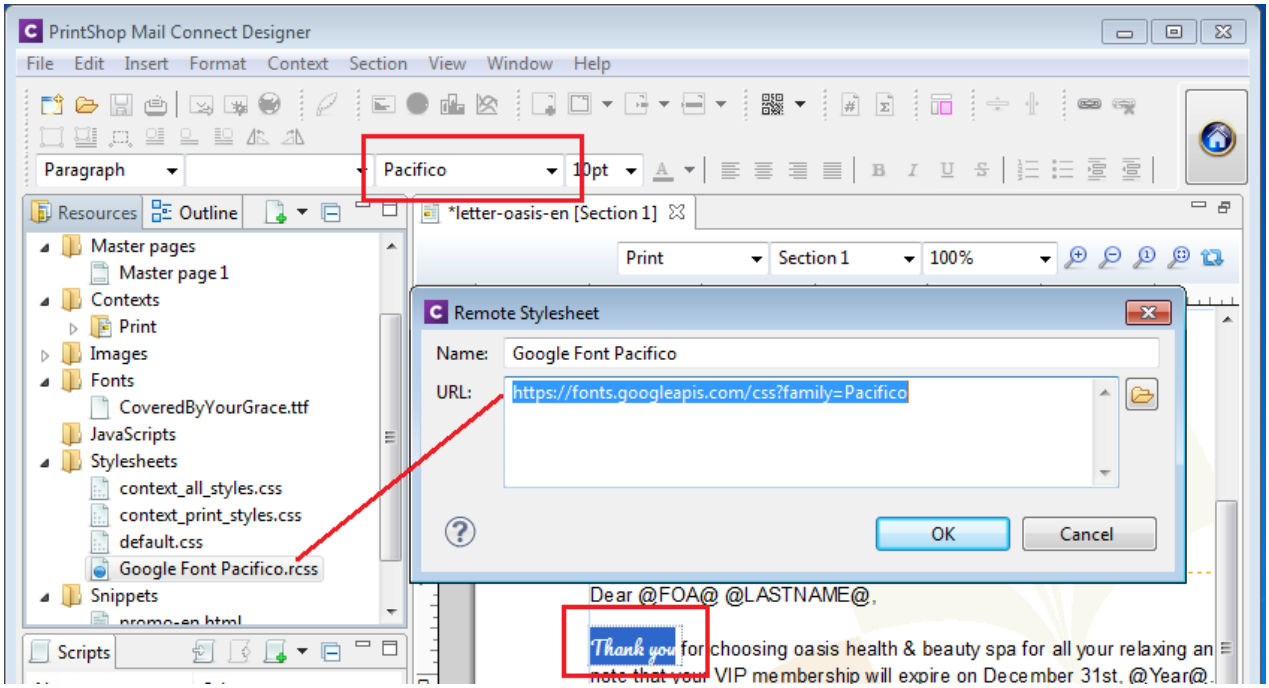
A wizard has been created to simplify the process of creating conditional print sections. Simply right mouse click a print section in the *Resource* panel and choose *Make Conditional…*
The wizard allows you to enter a basic condition based on a data field value and either *skip* or *print* the section. When the specified condition is *true* the selected *Action* is applied (which can be Print or Skip) otherwise the opposite is in effect. The icon of the section in the *Resources* panel identifies if the section is printed or skipped. Use the *Expand* option to reveal the underlying *Control* Script. (SHARED-47661)



Simplified Web Font support

Online font resources (such as Google Fonts) allow you to use their fonts for both commercial and non-commercial projects. Using these online fonts in your documents is a matter of linking to the Stylesheet file hosted by these services. Simply create a *Remote Stylesheet* entry and paste the location of Stylesheet file in the *URL* field. This will make the font available to the

application. It will be added to the Fonts menu and can be used in your custom CSS Stylesheets, via the `font-family` property. (SHARED-56637).



Dynamically set Media Background Images

Support has been added for dynamically setting the path of media backgrounds at run time (aka Virtual Stationery). This is achieved via the Control Script API. The path can be set to an image in the *Images* folder but also to a file on disk (the http:// and https:// protocols are not currently supported). This greatly simplifies template management in situations where a design is shared between different brands. This technique can also be used to dynamically set the stationery image for the preview template in Connect Send environments. (SHARED-53522)

```
var myMedia = merge.template.media["Media 1"];
myMedia.stationery.front.enabled = true;
myMedia.stationery.front.url = "file://C:/letterhead.pdf";
```

Page Breaks inside Lists

Support has been added to allow the splitting of lists across pages. This includes *Widows* and *Orphans* control for ordered (`<ol>`) and unordered (`<ul>`) lists.
The *Orphans* CSS property specifies the minimum number of lines at the bottom of a page and the *Widows* property specifies the minimum number of lines after the page break. To prevent

page breaks inside these elements simply add `page-break-inside: avoid;` to your stylesheets. A formatting dialog for these elements will be added in a future version. (SHARED-14092)

Installer improvements

Improved error capture, handling and messaging. (SHARED-40209)

Korean Language Support

PrintShop Mail Connect 1.8 is now available in Korean, in addition to the other languages already supported. (SHARED-40161)

Context Sensitive Help

Context Sensitive Help has been added to PrintShop Mail Connect. Selecting Help (via button, or `F1`) in dialog boxes or screens will now generally take you to the Help page most closely associated with the calling dialog box or screen. Context sensitivity will continue to be incrementally introduced and improved hereafter. (SHARED-45766)

Database connection deadlocks resolved

We have added a timeout period to all Connect back-end database connectors, as well as increasing the amount of database connection threads to better match the capabilities of the hardware. This greatly reduces the chances of database deadlocks or bottlenecks when processing jobs. (SHARED-57252)

Improved error handling of Merge Engine errors

In some rare circumstances XPCOM initializations would fail in Merge Engines. Thereafter the Merge Engine would continue to run but would be unable to process any further requests. Additional Merge engines would then be launched but the originals did not shut down, eventually leading to resource shortages and subsequent job failures. This issue has been addressed and XPCOM initialization errors in Merge Engines now cause the Merge engine to terminate and restart cleanly. (SHARED-57270)

Anchored positioned boxes losing style attributes

Absolute positioned (Anchored) elements would lose some style attributes under certain circumstances. These issues would only occur when the absolute positioned element had multiple style attributes that ended with the text "*top*" or "*left*". Such as is the case with

"*padding-top*" and "*top*". If both those attributes were set, then only one of the attributes would be retained. (SHARED-57361)

- Customers upgrading from 1.6.1 to 1.8 will not experience any issues with their templates.
- Customers upgrading from 1.7.1 or 1.7.2 to 1.8 will experience problems only if they have saved their templates within 1.7.1. or 1.7.2, and only if those templates contained absolute positioned objects with specific inline CSS styles that end with **top** or **left**, such as **padding-top**, **padding-left**, **border-top** etc.
  In that case those specific styles will be gone and they will either need to restore a backup from before 1.7.x of those templates or manually set the styles again in 1.8 and save the templates.

IPDS Data exception

In some instances IPDS output would fail, due to data records exceeding maximum limits and continuing in the subsequent record. This scenario is now catered for and the error is no longer encountered. (SHARED-57723)

Issues with 2D Barcodes on Nipson N7000 printers

2D Barcodes (such as DataMatrix) were corrupted when printing on Nipson N7000 printers. This error has been fixed. (SHARED-57767)

Changes made to Output Speeds in Connect 1.8

A speed throttling issue was discovered that allowed some users to exceed license limitations. This issue has been corrected, and output speeds will now more accurately reflect license speeds.

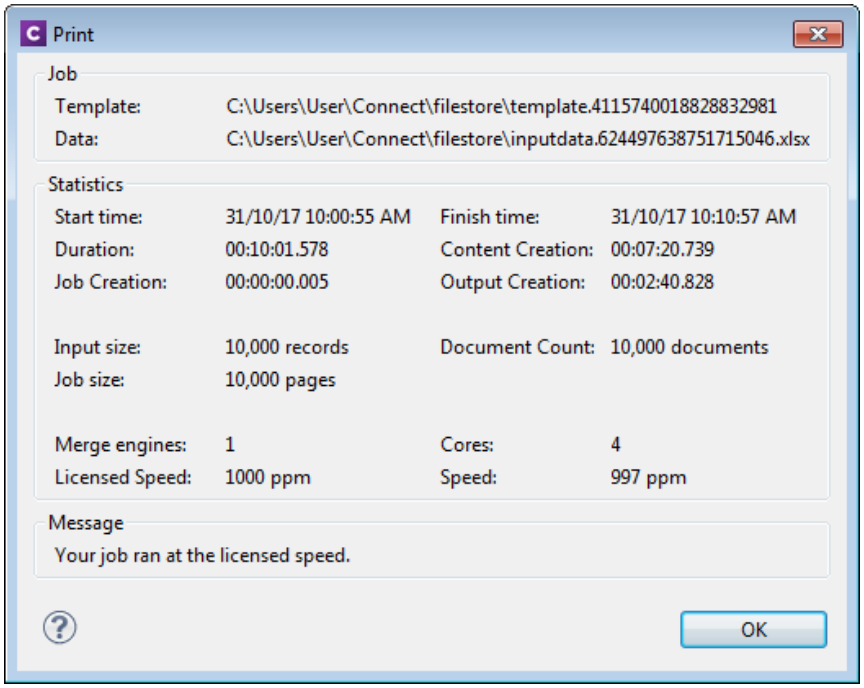
Connect 1.8 Performance Related Enhancements and Fixes

Faster printing in PrintShop Mail Connect

For PrintShop Mail Connect, the speed limit in the license will be applied differently, resulting in faster printing. Instead of applying the licensed speed only to the final step of the print process (the Output Creation), the speed limit is now applied to the *entire print process*. Thus the time required for Data Mapping, Content Creation and Job Creation are now taken into consideration as well. The final Output Creation step will now run as fast as required to have the entire print process completed at the licensed speed.

For example, if a 1000 page job needs 40 seconds for content creation, output creation (Weaver engine) is allowed to run at 3000 pages per minute to have the entire print process finish in 1 minute (assuming a licensed speed of 1000 pages per minute). Previously Output Creation would be limited to the licensed speed and cause the whole process to take 1 minute 40 seconds.

PrintShop Mail Connect now also shows a post printing report with details on job size and time spent. (SHARED-56386).



Connect 1.8 Designer Enhancements and Fixes

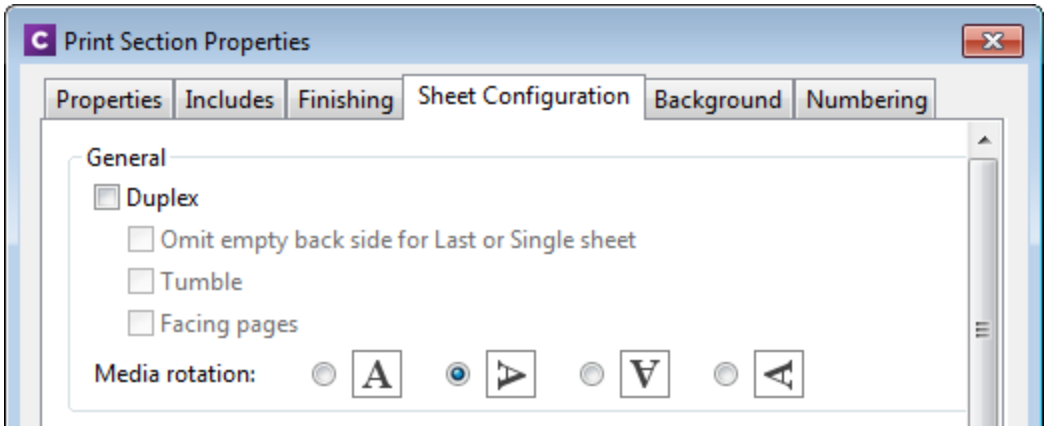
Automatically Fit Text to Container (Copy Fit)

The Designer can now automatically scale text content to fit the boundaries of a box (inline or absolute positioned `<div>`). Scaling text to fit a container is a very popular feature when creating personalized post cards and the like.

The option is found in the *Content* tab of the *Box* properties dialog and can be set to scale all text or a specific element in that box by entering a CSS selector. (SHARED-37702)

Rotate Print Sections individually

An option has been added to rotate individual *Print Section* orientations via the *Sheet Configuration* dialog (SHARED-46086)



Snap Guidelines to Ruler

Dragging a guide whilst holding the Shift key will snap the guideline to the closest mark on the ruler. (SHARED-54465)

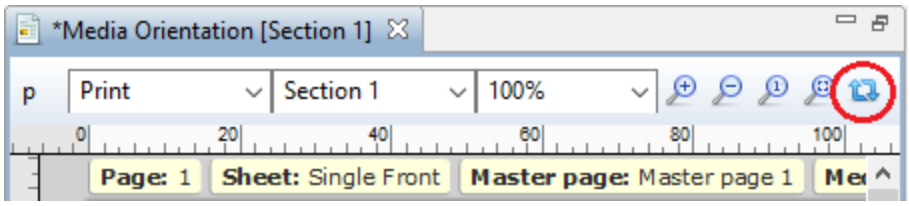
Toggle Comments On/Off via Shortcut Keys

Toggle comments off or on in HTML, CSS and JavaScript editors via a keyboard combination. Use `Ctrl + /` to comment out a single line and `Ctrl + Shift + /` to comment out multiple lines. (SHARED-56440)

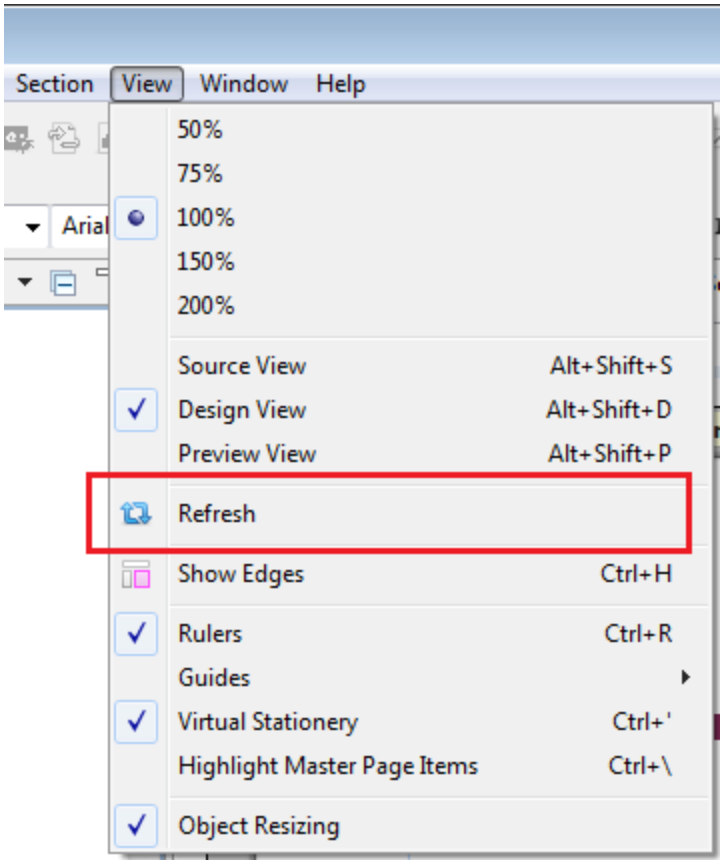
Refresh View button added

You can now refresh the contents of both Design and Preview views via the new Refresh button or new Refresh selection in the Menu. (SHARED-55616)

Button :

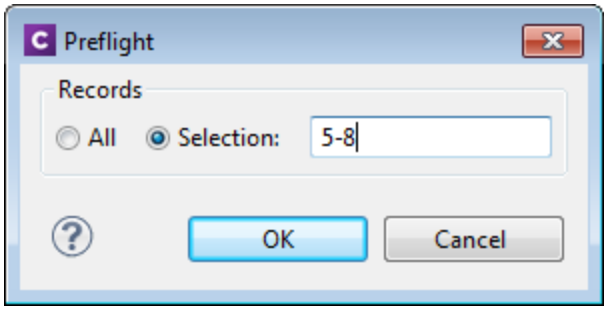


Menu selection :



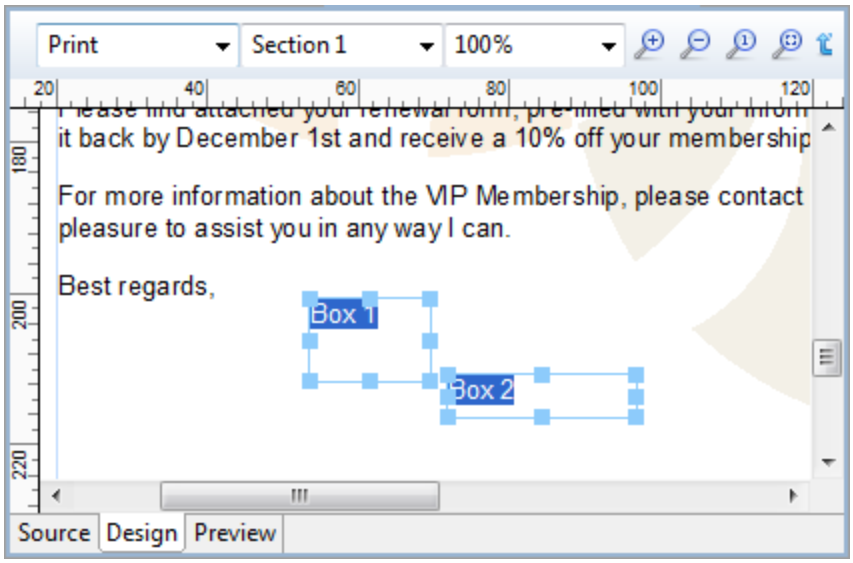
Specify Page Range for Preflight

You can now optionally perform Preflights on a range of records. (SHARED-35076)



Select and adjust multiple Box elements simultaneously

Multiple Box elements can now be selected at the same time.



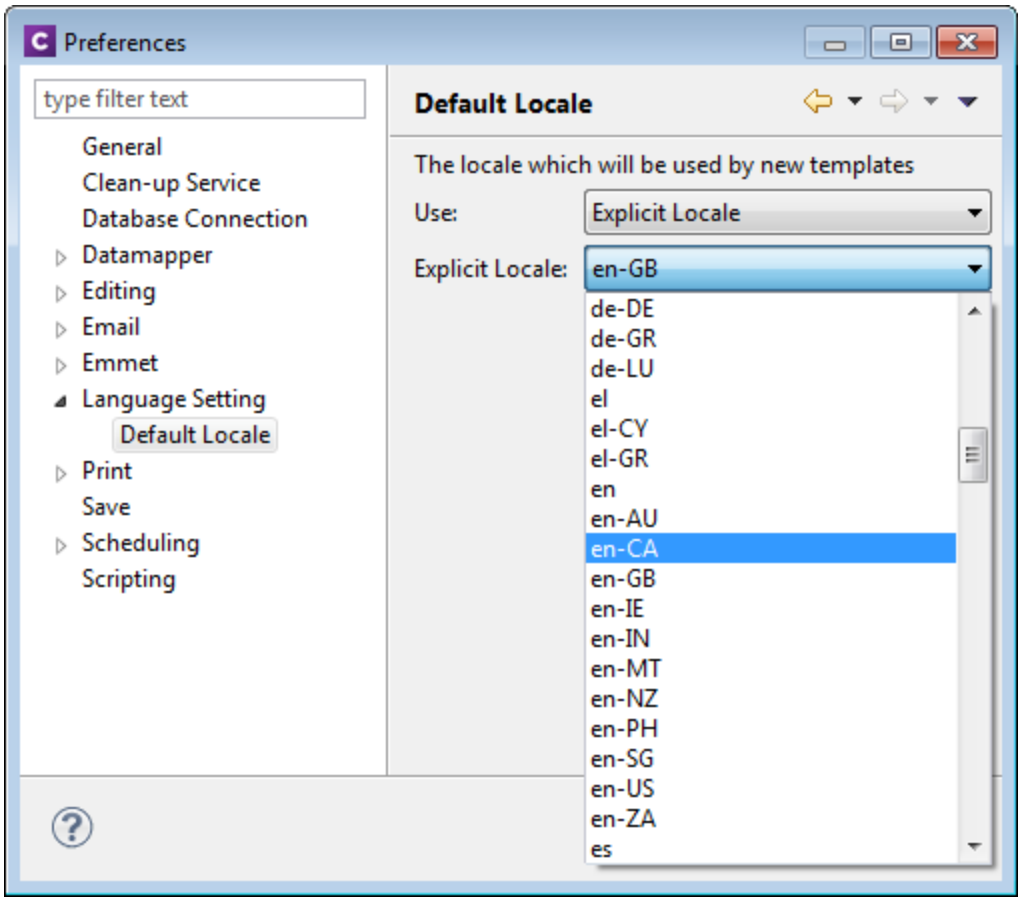
Once selected you can either move or resize the selected boxes as a group. You can move them either via the mouse, or by nudging them around a single pixel at a time with the arrow keys. When nudging, the boxes will not snap to guides. Otherwise, when moving or resizing multiple boxes, the box that was originally clicked (the reference box) will snap to guides if *Snap to Guides* is enabled. (SHARED-55636)

Specify/change name of Email Attachments

Ability added to overwrite the file name for email attachments through scripting. (SHARED-57120)

Set Template Locale

Previously Connect always assigned the *System Locale* to new templates. A new option has been added to the preferences to allow the selection of a specific locale. This selection will then apply to all new templates thereafter. It applies to Date and Time fields plus numeric and currency data fields. For example, a monetary data value in France (locale fr-FR) would apply the € (Euro) currency symbol and use the ',' (comma) as the decimal separator, whilst the same monetary data value in the US locale would apply the $ (Dollar) symbol, and use the '.' (full stop) as the decimal separator. (SHARED-56791)



Locales can still be changed in individual templates by the **Edit > Locale ...** option.

General Designer improvements

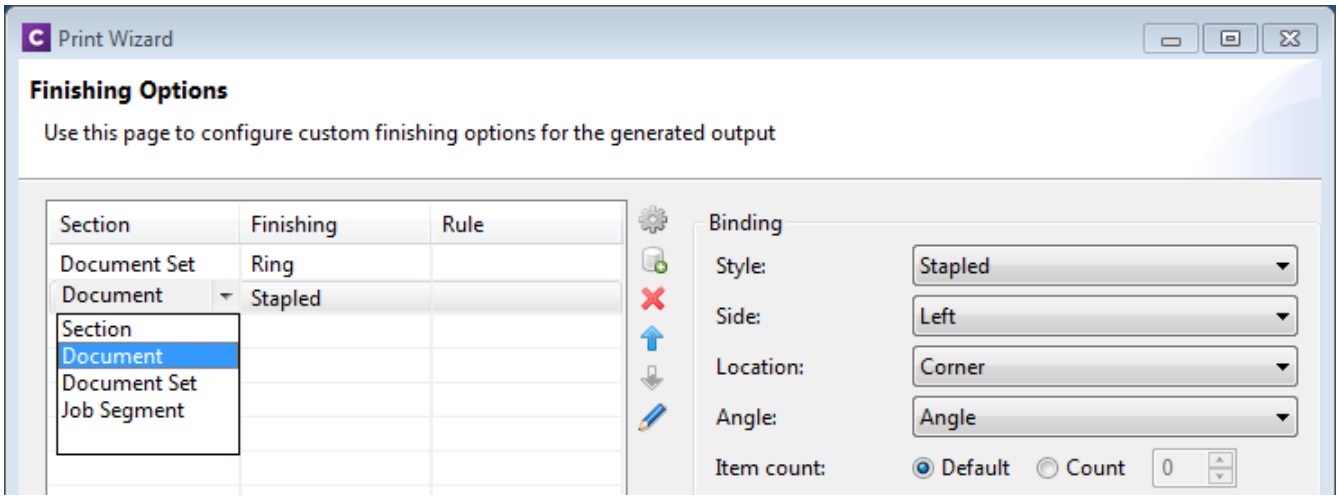
- **GS1 Datamatrix** barcode now supported. (SHARED-55999)
- Section, Media, and Master resources can now be duplicated by copy-pasting. (SHARED-52261)
- **Reopening a template** will put the focus on the section that was active when the template was closed. (SHARED-53199)
- **Keyboard shortcuts** to increase (`Ctrl + Shift + >`) or decrease (`Ctrl + Shift + <`) text size now work as expected. (SHARED-11660)
- "Problem" view renamed more accurately as "**Preflight Result**". (SHARED-56343)
- Added "**folding**" support for CSS editors and source editors for standalone HTML files. (SHARED-10717)
- Multiple values now allowed in the **Make Conditional** script wizard. (SHARED-57029)
- **Default colour** swatches can now be edited or overwritten. (SHARED-53670)
- Minor Designer interface inconsistencies fixed. (SHARED-54114)

Connect 1.8 Output Enhancements and Fixes

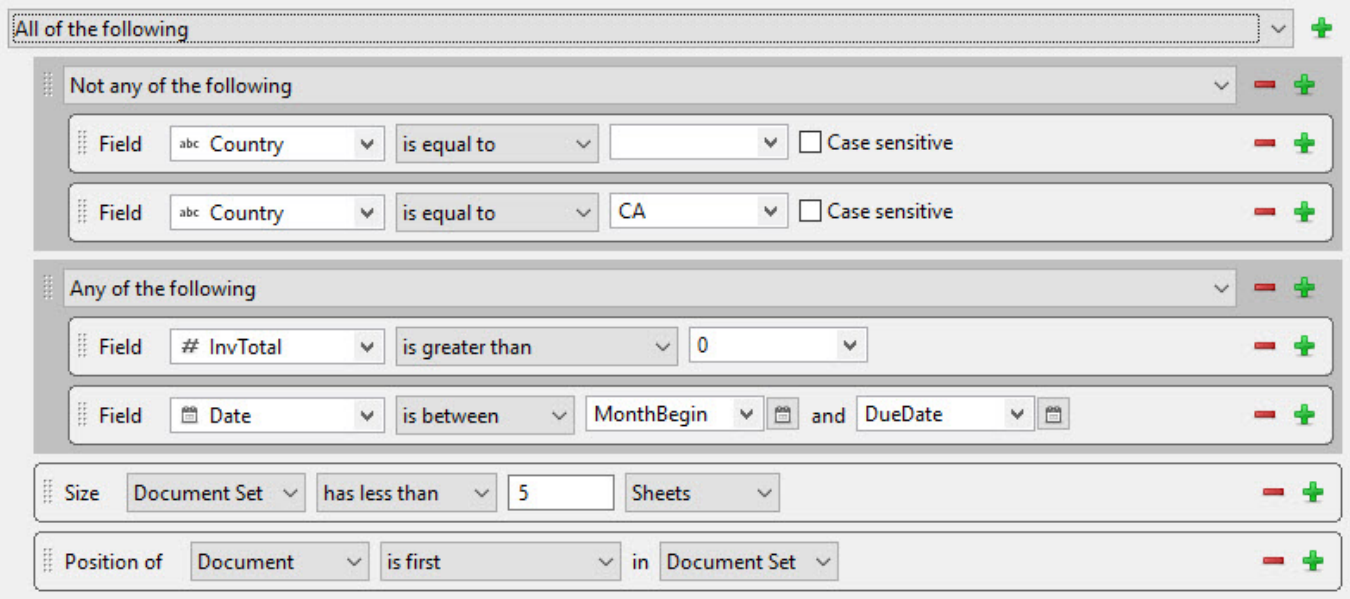
Dynamic Finishing

Print Output Finishing has been improved considerably, and is now much more powerful and flexible. In Templates, you could already set finishing for documents and sections. Job Creation would allow you to specify a different kind of Finishing for your documents and Templates. This has been extended to allow Finishing settings on all levels of Job Creation: **Document Sets**, **Job Segments** and **Jobs**. (SHARED-53277)

So now you can staple document sets and punch holes in your segments. You can also have multiple finishing settings at the same level.


Print Wizard
Finishing Options
Use this page to configure custom finishing options for the generated output
Section
Finishing
Rule
Document Set
Ring
Document
Stapled
Section
Document
Document Set
Job Segment
Binding
Style:
Stapled
Side:
Left
Location:
Corner
Angle:
Angle
Item count:
Default
Count
0

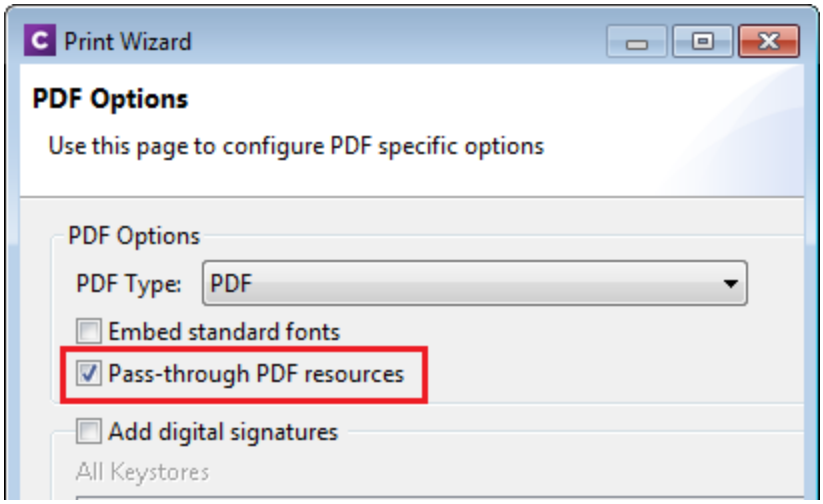
The reason we call this feature ***dynamic*** finishing, is that it includes a brand new rules editor to allow you to choose when to apply a finishing setting:



PDF Pass-through

Connect's output creation (Weaver engine) tries to write content the best way possible, depending on the chosen output format and optimization settings. However, there are cases when this might not be desired, such as when the graphics have already been optimized for the device and you do not want the software to change them.

It is now possible to instruct output creation to include PDF resources in the output file *as-is*. When used, it guarantees that the fidelity of PDF graphics used in a template is retained in the output. The resulting output will be less optimized, typically producing somewhat larger files. This option can also be useful when the output is showing unexpected results or to prevent rasterization of PDF output. Will this feature trigger visible differences in the output? No, in most cases not, but when printing highly optimized graphics, expect to see a slight difference in the printed output. This is only possible when PDF content is allowed in the output, meaning it can be used with PDF, IPDS and AFP output with PDF containers. For PDF output, this feature can be found in the Output Preset or Print Wizard page under **PDF Options**. (SHARED-56412)



For AFP and IPDS with PDF containers, this option needs to be set via a Printer Definition file. Printer Definition files for AFP and IPDS that have PDF pass-through enabled will be made available for download, and will be accessible from the Connect Welcome page.

Please note that when using a PDF from a Data Mapping in combination with Virtual Stationery, the PDF is not passed through when selecting this option in Connect 1.8. This is a known issue and will be addressed in a later release of OL Connect.

A second issue when using a PDF background via the Datamapper is that the resultant PDF output file *may* contain invalid font resources. Whilst the output can be viewed in Adobe

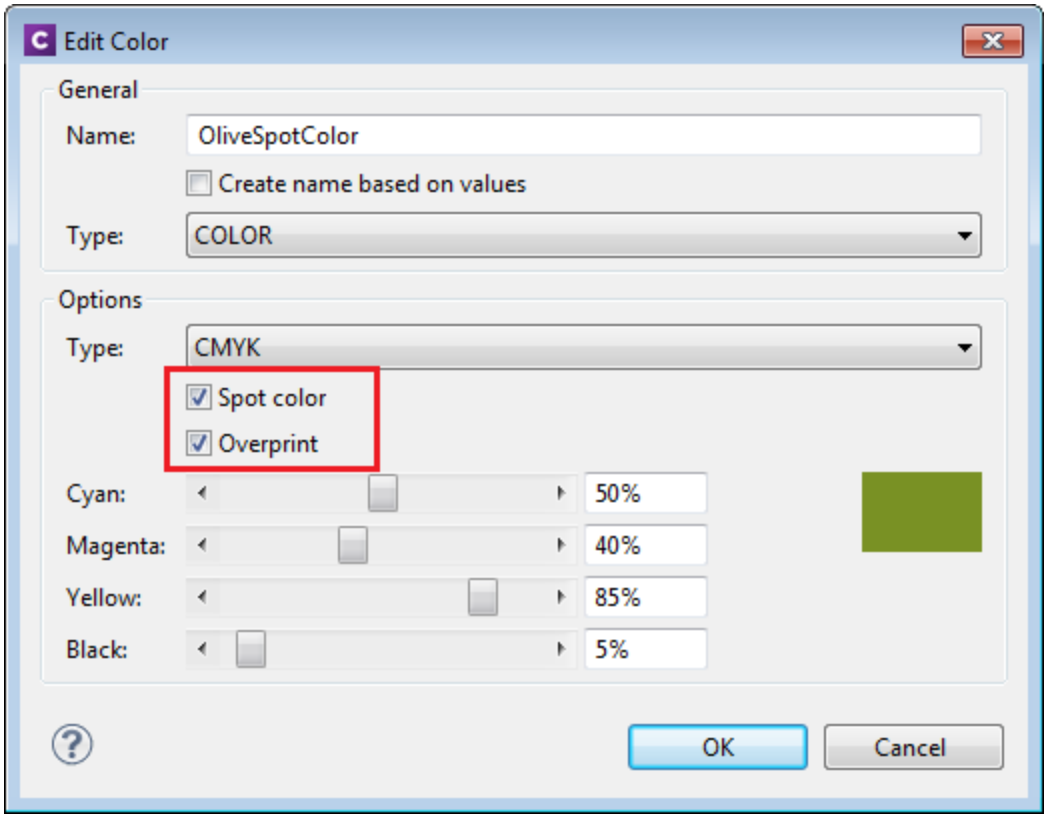
Acrobat Reader without issue and will print correctly on many printers, it will prompt warnings in Adobe Acrobat Professional's Preflight report and it should not be used as input for Connect Data Mapping. We recommend testing the output on your specific printer(s) to best determine whether this will be an issue on your specific printer(s) or not. This issue will also be addressed in a later release.

Overprint for Spot Colours

Overprinting certain content on top of other content is sometimes required. (SHARED-56743) For example:

- To deal with special print applications, such as applying special (invisible) inks that are intended to go on top of coloured areas, for instance printing UV ink or applying varnish to a certain area.
- To avoid mis-registration when printing black on top of coloured areas.

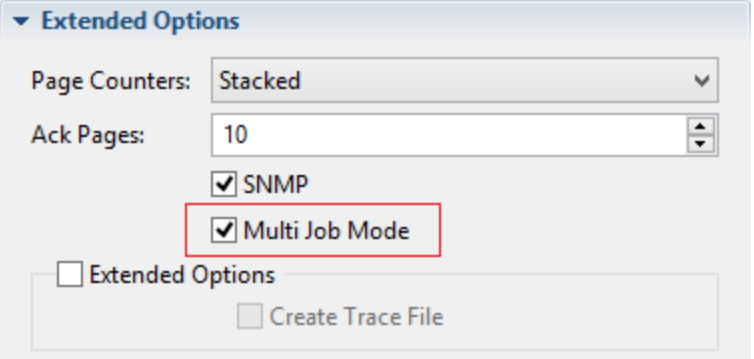
To support scenarios like these, Connect now supports Overprint for Spot Colours.



Note: Overprinting does not show on-screen in the Designer.

Print Output

- **Mixplex support added**
  An option has been introduced to omit empty back sides for the *Single* and *Last* sheet positions when Duplex is enabled, resulting in mixplex output. This helps in reducing costs in printing environments where page count or click-charging is applied. (SHARED-46965\55459)
- **Improved logging** of output generation. (SHARED-53367)
- **Continuous Printing on IPDS printers now supported**
  Option added to the Add/Edit Printer dialog, to provide support for continuous printing on IPDS printers.



For further details, please see **"Continuous Printing on IPDS Printers" on the facing page**.

Connect 1.8 Print Manager Enhancements and Fixes

Continuous Printing on IPDS Printers

When printing on high volume IPDS printers, it is best if the printer does not pause often. On continuous feed printers in particular, a pause can lead to malfunctions as well as cause delays. It's not just the pause itself, but also the additional time spent whilst the printer slows down and then speeds up again after the pause. The faster the printer, the bigger an issue this becomes.

The Connect Print Manager itself was increasing the pause time as it had to make sure that the currently printing job was completed before sending a new job.

To reduce the need for such pauses, support for **IPDS output incorporating multipage PDF containers** was added in Connect 1.8. Most (if not all) modern IPDS printers support Multipage PDF containers, which can be used in such a way to as remove pauses between print jobs. This means high-end printers can now be prevented from pausing.

To cater for IPDS output with Multipage PDF containers, the Print Manager has been significantly updated.

For further information about this, please see the Print Manager section in the online Help. (SHARED-56881)

General Print Manager improvements and fixes

- The entire row is now highlighted when selected in the SNMP printer status view of the Print Manager. (SHARED-41114)
- Can now drag print jobs back and forth between the Printer Queue and the Job Queue.
- A multitude of small GUI improvements were made, including catering for new continuous IPDS printing.

Known Issues

Installation Paths with Multi-Byte Characters

When installing the Traditional Chinese or Japanese versions of Connect, if the user specifies an alternative installation path containing multi-byte/wide-char characters it can break some of the links to the Connect-related shortcuts in the Start Menu and cause an error to appear at the end of the installer. The workaround for the moment is to use the default installation path. The problem will be addressed in a later release.

Switching Languages

Changing the language using the **Window>Preferences>Language Setting** menu option does not currently change all of the strings in the application to the selected language. This is a known issue and will be fixed in a later release.

In the meantime we offer the following workaround for anyone who needs to change the language:

1. Go to the .ini files for the Designer and Server Config:
    - C:\Program Files\Objectif Lune\OL Connect\Connect Designer\Designer.ini
    - C:\Program Files\Objectif Lune\OL Connect\Connect Server Configuration\ServerConfig.ini
2. Change the language parameter to the required one under Duser.language=en | es | de | fr | it | ja | ko | pt | tw | zh

Only one of the above language tags should be selected. Once saved, Connect will appear in the selected language at next start-up.

GoDaddy Certificates

When installing Connect offline, dialogs allow installing the GoDaddy certificates. Most users should use the default settings and click **Next**. In some cases, however, this may not work correctly. For this reason those users should activate **Place all certificates in the following store** and then select the **Trusted Root Certification Authorities** as the target certificate store.

Available Printer Models

Note that only the single Printer Model (Generic PDF) will appear on the **Advanced** page of the **Print Wizard** by default.

To add additional printer models click on the settings button next to the Model selection entry box.

Colour Model in Stylesheets

The colour model of colours defined in a stylesheet can sometimes change after editing the stylesheet. This is a known issue and will be addressed in a subsequent release.

Image Preview in Designer

If in the Windows Internet settings (**Connection Settings > LAN configuration**) a proxy is enabled, but "Bypass proxy settings for local addresses" is not checked, the image preview service, conversion service and live preview tab in the Designer will not work and exhibit the following issues:

- Images will be shown as 0 size boxes (no red 'X' is displayed)
- Live preview does not progress, and when re-activated reports "browsers is busy"

To fix the issue you must check the "Bypass proxy settings for local addresses" option.

VIPP Output

Some templates set up with landscape orientation are being produced as portrait in VIPP. It can also sometimes be the case that text and images can be slightly displaced. These are known issues and will be addressed in a later release of Connect.

PrintShop Mail Connect Release Notes 1.7.1

Overview

This document provides an overview of the new features and enhancements in PrintShop Mail Connect 1.7.1.

Installing PrintShop Mail Connect 1.7.1

- PrintShop Mail Connect is released as a 64 Bit version only.
- Full details on installing and licensing PrintShop Mail Connect can be found in the online help in the installer.
- Note that PrintShop Mail comes with a 30 day trial licenses by default.

Connect 1.7.1 General Enhancements and Fixes

Template Reports added to Connect

Generate a report in PDF format containing the most important information about your template. The report lists contexts, sections, master pages, scripts, graphic files, and any other resources used, along with their properties. This report can be added as part of your project documentation.

The report is created using Connect technology, and it generates an XML file and thumbnails, allowing you to create your own custom report structure and corporate styling. This can be achieved by altering the underlying Connect Template and DataMapper configuration.

The following screen-shot shows an extract from a sample report that was created for the *OL Mobile Letter* template:

Print Context

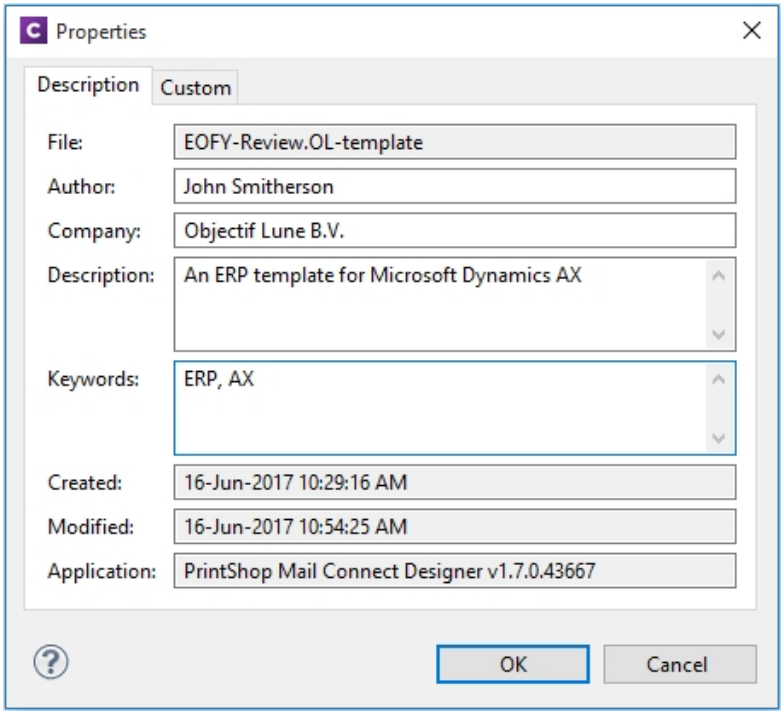
Section 1

Properties	
Name	Section 1
Page size	Letter
Dimensions	8.5in x 11in
Orientation	portrait
Margins	**Top**: 1.5in, **Right**: 0.75in, **Bottom**: 0.75in, **Left**: 0.75in
Bleed	**Top**: 0in, **Right**: 0in, **Bottom**: 0in, **Left**: 0in

Includes	Type	Location
context_all_styles.css	css	css/context_all_styles.css
context_print_styles.css	css	css/context_print_styles.css
default.css	css	css/default.css

Document Properties

Document Properties can now be added to Templates. This allows you to specify properties such as the document author, the customer name and other important references. You can also add custom key/value pairs. The respective properties can be retrieved in scripting and are thus available as content in your documents. The information is also included in the *Template Report* feature. (SHARED-47780)



Stability improvements

- Improvements made to the Clean-up service. In some production environments the database Clean-up could not keep pace with database growth, leading to the database gradually filling up. This has been fixed through an improved internal database structure and more efficient queries and deletions. (SHARED-46465/52345)

"Enhance with Connect" option added for PDF files in Windows Explorer

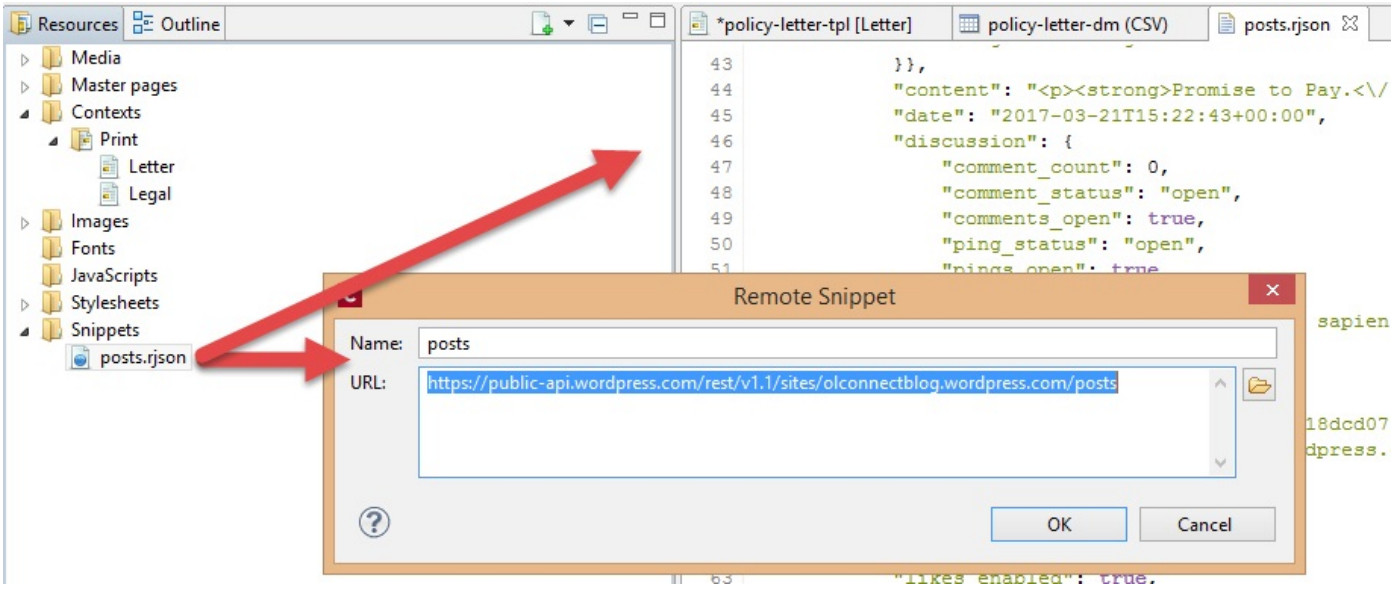
A Windows Explorer context menu entry "**Enhance with Connect**" has been added for PDF files. When a user selects this context menu entry, PrintShop Mail Connect Designer opens with a prefabricated template, that uses the selected PDF file as the background. (SHARED-15350/47156)

Support added for Remote HTML and JSON Snippets

In PrintShop Mail Connect 1.7.1 we introduce the concept of remote snippet resources. These snippet entries have a Name and URL property (e.g., the hyperlink to the endpoint) and reside in the Snippets folder located in the Resources panel.

In scripts these snippet entries are referenced just like regular snippets, e.g., `loadhtml ('snippets/my_content.rhtml')` or `loadjson('snippets/posts.rjson')`. Note the "r" in the file extension.

Having the snippet entry in the **Snippets** folder within the **Resources** folder allows us the simplest overview of the resources used. In previous versions this behaviour would have had to be captured in script and therefore would not have been directly visible as part of the resources. This new approach greatly simplifies maintenance of the URL, as it can now simply be updated in the Resources panel rather than by browsing through all the scripts. (SHARED-42314/52591)

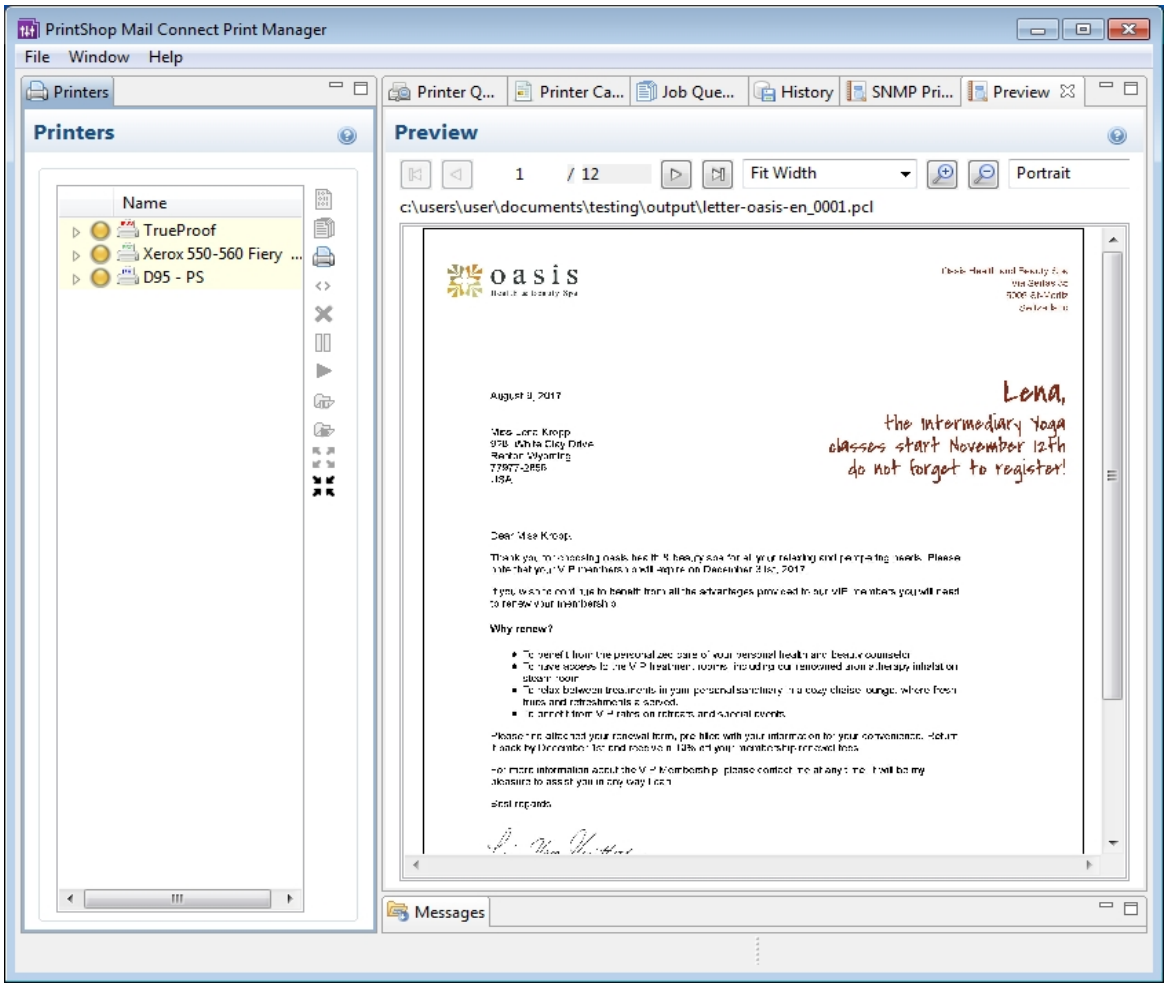


Installer improvements

- The PrintShop Mail Connect 1.7.0 installation did not work on machines running Windows 10 build 1703 (i.e. the "Creators Update", released March 2017). This has been fixed for PrintShop Mail Connect 1.7.1. (SHARED-56800)
- The **silent installation** process has been enhanced, and now supports the following:
  - **Setting the repository**. This can be configured via the "*product.repository*" entry in "*install.properties*". (SHARED-17841)
  - **Selecting a dedicated locale** (language and country code) for the Connect applications. These can be configured by the "*user.language*" and "*user.country*" entries. (SHARED-18381)
  - Improved reporting of Silent Installation success or failure. (SHARED-17723)
- Microsoft SQL server connection settings added to Connect Installer. (SHARED-36866)
- The Update Client will now run the installation in the same language as the original installation. (SHARED-37868)
- Installer has been made more robust, and will now continue (with warning messages, if applicable) when it encounters any of the following scenarios:
  - If Server start-up was unsuccessful during installation. (SHARED-39398/46837)
  - If no Database connection could be established. (SHARED-39400)
  - The Installer now checks if the OL Connect MySQL service is in the proper state and resident in the expected folder. If is not, instructional warning messages are now displayed. (SHARED-40309/45431)
  - If Connect folders that should have been deleted were found upon re-installation. (SHARED-41420)

Print Manager improvements

- A document preview has been added to the Print Manager, for IPDS, PCL and PostScript documents. (SHARED-43959)



NOTE: The Postscript viewer requires manual installation of the free Ghostscript application.

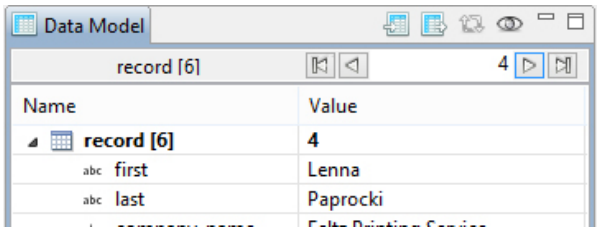
Connect 1.7.1 Designer Enhancements and Fixes

Edit and Save CSS, HTML, JavaScript and JSON files within the Designer

Ever needed to quickly edit an external CSS, HTML, JavaScript or JSON file? The PrintShop Mail Connect 1.7.1 Designer now allows you to open and save these file types via the **File** menu. (SHARED-42094)

Data Model Panel Enhancements

Various enhancements have been made to the **Data Model** panel. The browse options of the main record are now *sticky* and do not move out of view when working with a large number of data fields. An eye icon has been added to the toolbar, and is used to toggle the visibility of the *ExtraData* field. In addition, you can select and group multiple fields in order to collapse them out of view (and expand them back, obviously), which is particularly useful when dealing with large data models that force you to constantly scroll up and down to bring a specific field into view. (SHARED-45370/54106)



The Data Model panel has also been enhanced to allow alphabetical sorting of detail tables. (SHARED-47169)

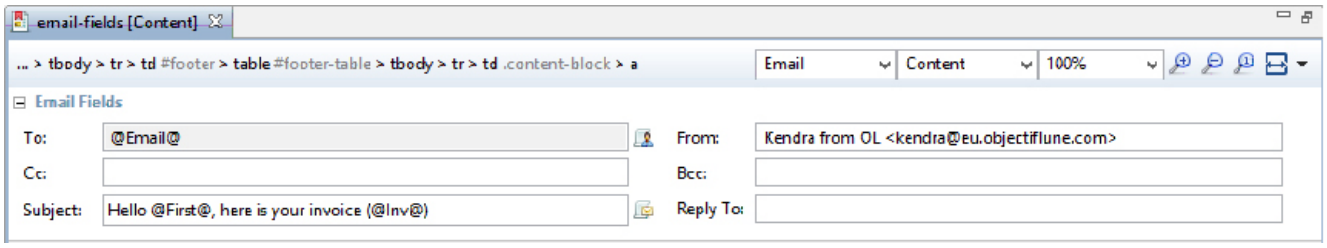
Simpler Invocation of Email Script Wizards

Invoke email related script wizards simply by clicking the labels in the email information bar. (SHARED-47329)

Simplified Email fields User Interface

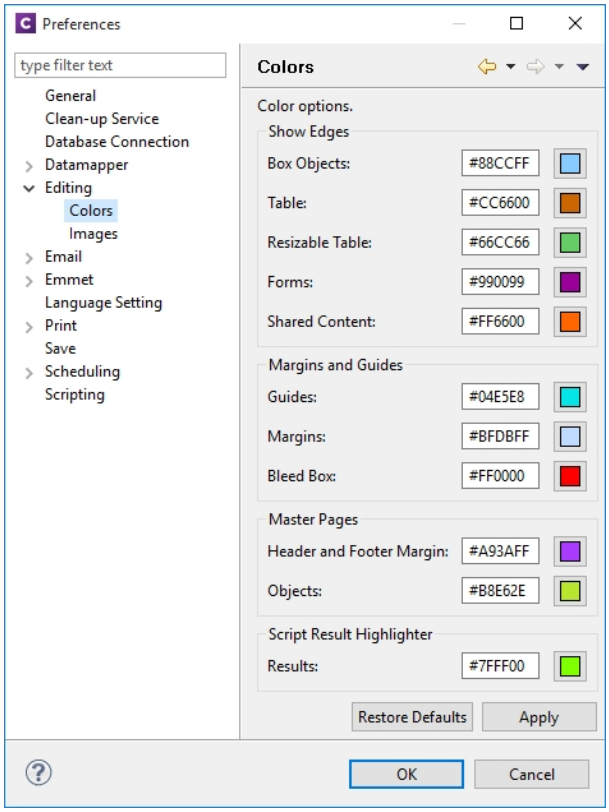
Create *To*, *CC*, *BCC*, *From* and *Reply To* email scripts by dragging and dropping a data field to the respective input field or type a static address directly in the input fields. (SHARED-9178)

Type the subject in the Subject email field and drag and drop data field(s) to positions in that string to make a personalized email subject without any scripting. (SHARED-51475)



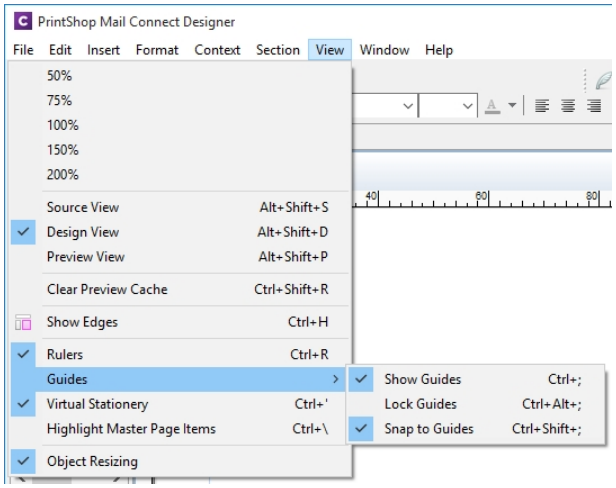
Improved Customization of the Designer interface

Customize your interface by selecting your own colours for object edges, margins, guidelines, etc. (SHARED-49841)



Guideline behaviour improved

Along with visible/invisible settings, Guidelines can now be locked in place or set to snap to objects, using the new **Guides** option in the View menu. (SHARED-47159).



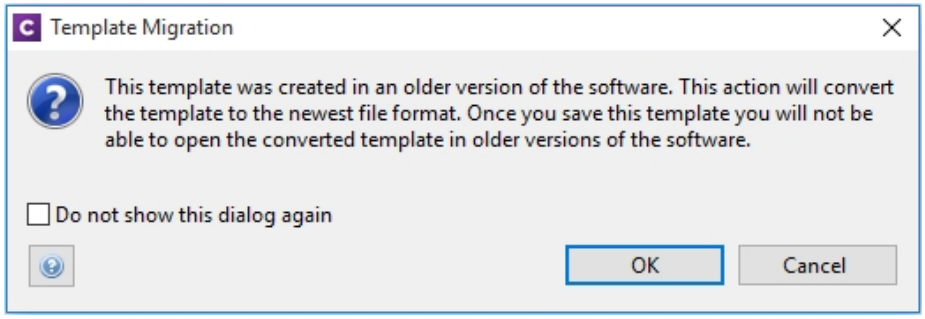
Warning now displayed when opening templates created in an older version

When PrintShop Mail Connect opens an older template file it is automatically migrated to the template structure of the current version. Saving the file in the new version would thus update the file format and prevent the document opening in an older version.

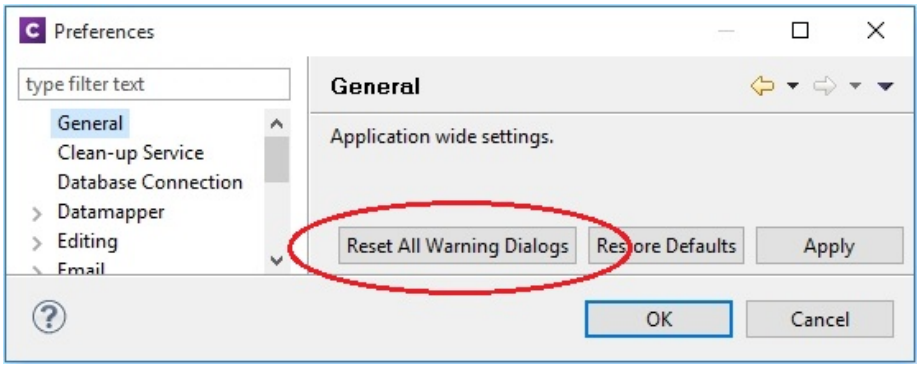
A warning is now shown when opening a Template created in an older PrintShop Mail Connect version, allowing you the chance to save the Template to a new file, leaving the original intact. (SHARED-51912)

Turn Warning dialogs Off/On

A "**Do not show this warning again**" check-box option has been added to many PrintShop Mail Connect Warning dialog boxes:

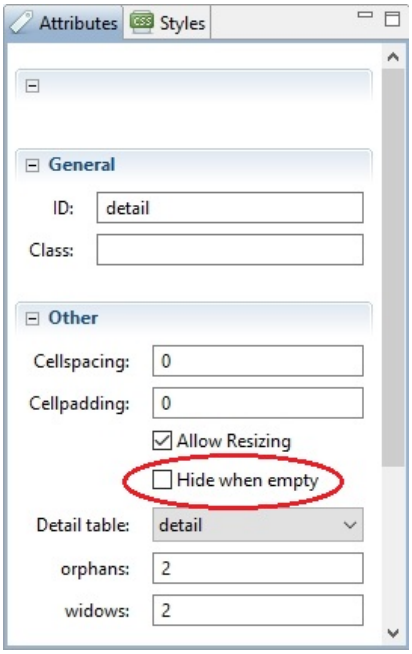


These Warnings can be switched on again at any time thereafter, via the "**Reset All Warning Dialogs**" button in the General Preferences dialog. (SHARED-16962)



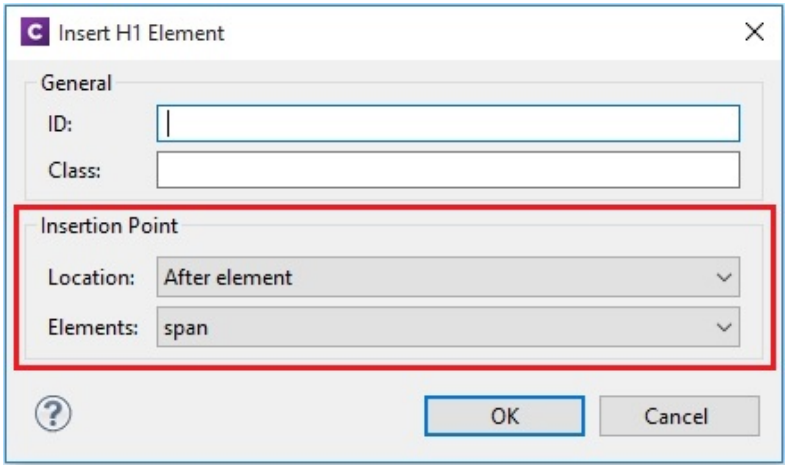
Option to automatically Delete a dynamic table when the table is empty

An option has been added to allow you to automatically delete a dynamic table when the data table is empty. To do so, select the entire table, and then tick the "**Hide when empty**" checkbox in the Attributes panel. (SHARED-43537)



Replace elements with data-insert-location when inserting HTML elements

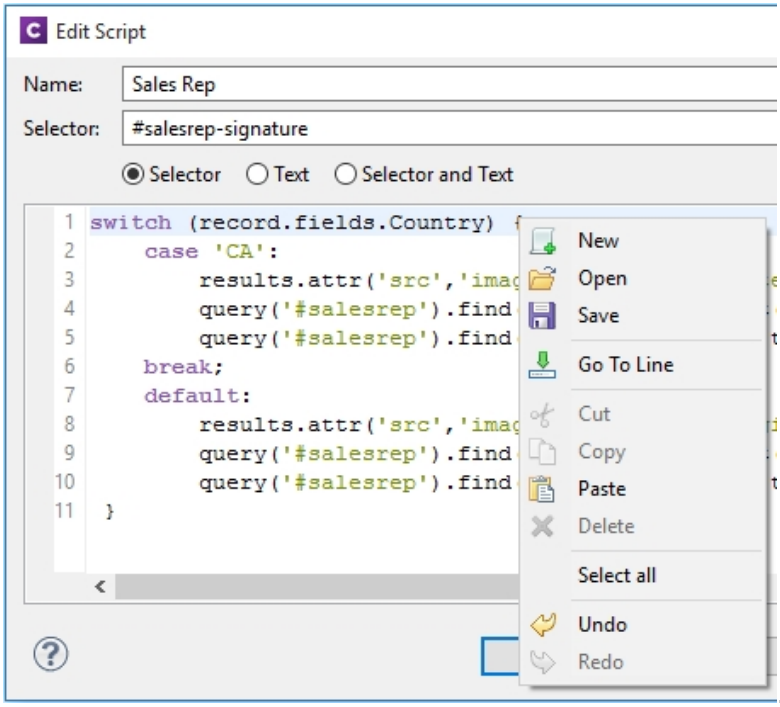
When inserting an element from an **Insert dialog**, Connect now checks the *data-insert-placeholder* attribute. The value of the attribute is then used to set the default value for the *Insert Location* option within the Insert dialog. If the attribute is not found, things behave as in previous versions.



This ticket also introduces the Replace option for the Insert Location drop down. When selected the to-be-inserted element(s) will replace the currently selected element(s). (SHARED-52369)

Scripting improvements

- Context menu added to the **Edit Script** dialog. (SHARED-45381)



- **Find and Replace** functionality has been added to Script editors. (SHARED-48424)
- New menu option to **rename** Scripts or Folders has been added to the Context Menu within the Script panel. (SHARED-48607)
- Support added for **copy and paste** of folders and scripts within the Scripts panel. (SHARED-49299)
- The JavaScript **parseInt()** method now defaults to using base 10 arithmetic rather than base 8, as defined in the ECMASCRIPT specifications. (SHARED-49010)

General Designer improvements

- **Duplicate and Delete line(s)** using shortcuts in the Stylesheet, JavaScript and HTML editors. Use `Ctrl+D` to **duplicate** and `Ctrl+Shift+D` to **delete** the currently selected lines. (SHARED-46928)
- Entering **geometry values** without stating a specific unit type will now automatically assign the default unit type to the entry. (SHARED-50656)
- When **deleting an element** (such as a Barcode or a Chart) on a page, a check will now be made for associated scripts. If any are found, the deletion step will provide an option to delete those scripts as well. (SHARED-45675)
- An "**All Files (*.*)**" filter was added to Save/Save-as dialogs. (SHARED-28237)
- **Icons** have been added for JS and CSS files in the Includes dialog to make it easier to distinguish between local and remote resources. (SHARED-47936)
- **Abs box** grippers and borders now display at a consistent thickness regardless of zoom level. (SHARED-50175)
- Support added for **dynamically setting the media background image** and its options via a Control Script. This only works for PDF files residing in the template at the moment. (SHARED-53524)

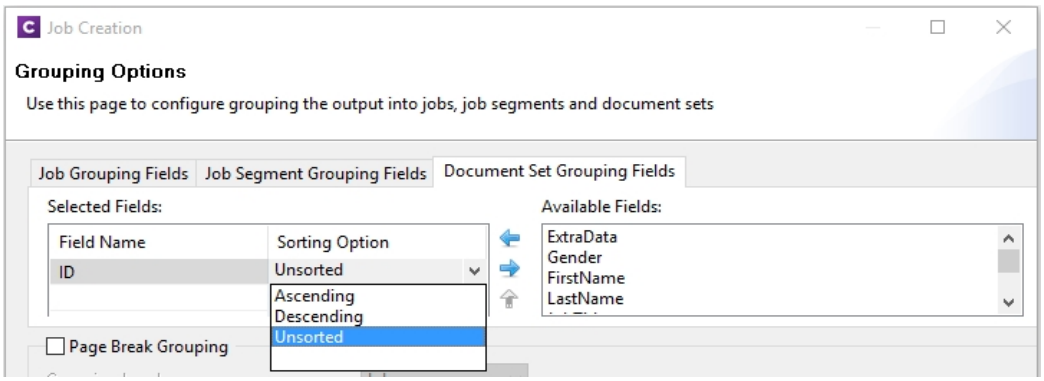
Connect 1.7.1 Output Enhancements and Fixes

Grouping With and Without Sorting

Sometimes the data used for generating documents is already pre-sorted, but you may still need to group documents into sets or segments. In those cases, the grouping process should not reorder the documents. This has now been implemented in PrintShop Mail Connect 1.7.1.

Consider the following example: data has been pre-sorted for postal sorting, which means that documents for the same customer will also be in consecutive order in the job (assuming a customer has a single postal address). If we want documents for the same customer to go into the same job, we can use grouping to create document sets and we might use the customer number for this grouping. When the customer number changes, we want a new document set to begin. If grouping by customer number also sorts by customer number, our pre-sorted order will get messed up.

The Job Creation settings have been improved to allow this kind of grouping. Sorting ascending, descending or not sorting at all can be set per field used for grouping. (SHARED-45125)



This means that, apart from straightforward cases where we are grouping with or without sorting, it is also possible to create combinations where some fields do alter the sort order and others have no effect.

Please note that grouping without sorting also means that any documents that have the same value for the same grouping field (i.e., customer number in the example above), but which are not consecutive in the input data, will not end up in the same group.

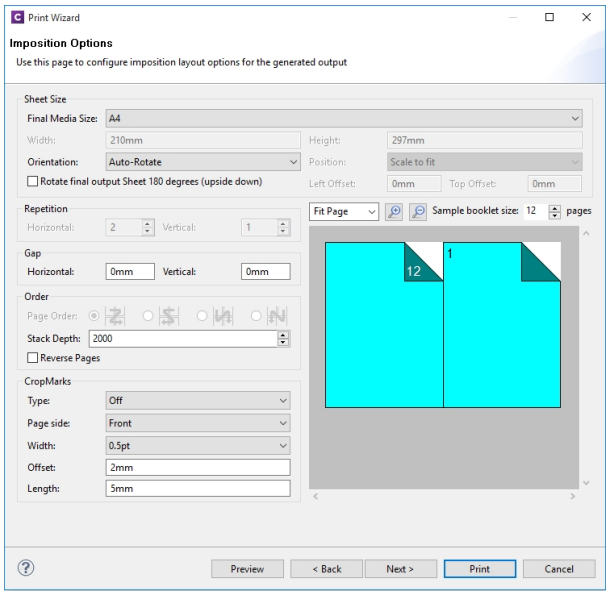
The settings for **Grouping** are available both in the **Job Creation Settings** dialog and the **Advanced** mode of the **Print Wizard**.

Progress of External Sort now displayed

When using an external sort, there was no feedback about how the external sort program was progressing. A new dialog control has now been added to Connect which displays the progress of the external sort in real time. (SHARED-53601)

Improvements made to Imposition Options dialog

The settings page for Cut and Stack Impositioning has been improved to show a sample of the chosen imposition settings.



Additionally, some settings on the Imposition Options page affect the way that booklets are created. These settings are now editable, so settings such as the gap between pages can now also be set for booklets. (SHARED-31097)

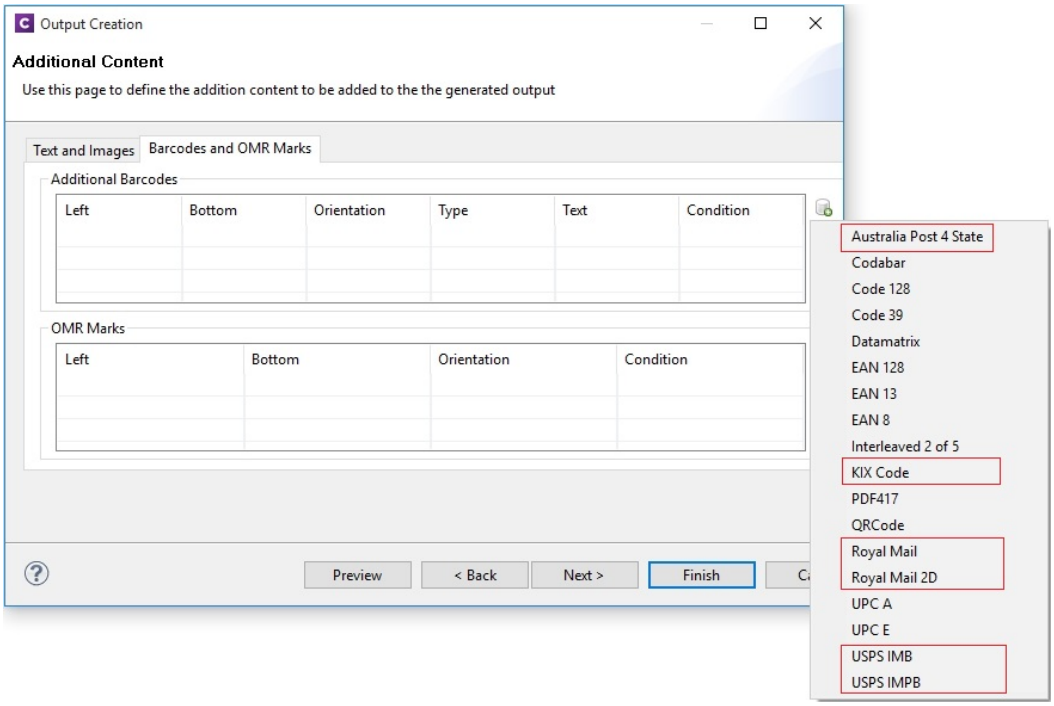
Additional Postal Services Barcodes added to Output Creation

Barcodes for postal services are excellent candidates for adding during the Output Creation steps, rather than during Content Creation.

Reasons for this include:

- They often cannot be added during Content Creation because they depend on document size (or weight) and on a sort order that is determined during Job Creation.
- They need to go in a fixed position, dependent upon the envelope window, rather than document design.
- It can be desirable to have templates independent from the postal service doing the delivery, in cases where there is a choice between postal services. This makes it relatively easy to switch to whichever service is offering the better rates.

To support these scenarios better, a number of postal service specific barcodes have been added to Output Creation, in addition to Content Creation. (SHARED-54755/54962/55046)



The new barcodes include:

- Australia Post 4 State
- KIX Code (Dutch postal service - Post.NL)
- Royal Mail (UK)
- Royal Mail 2D (UK)
- USPS IMB (US)
- USPS IMPB (US)

Some of these barcodes have specific requirements in order for them to be usable. The respective postal services provide specifications and sometimes also the tools for generating the content of these barcodes.

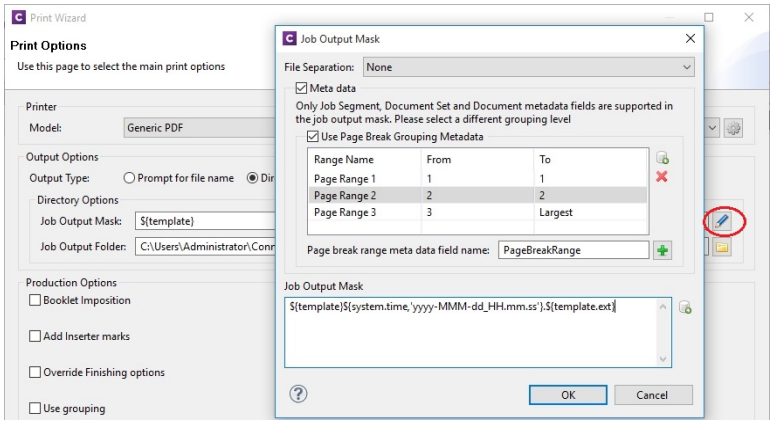
The checksums needed for Australia Post 4 State and IMPB are calculated automatically.

Fixed issue with Merge Engine memory usage

The Merge Engine would slow down when running some jobs that used external JavaScript files. These memory issues have now been resolved. (SHARED-47242)

Job Output Mask improvements, to simplify working with output file names

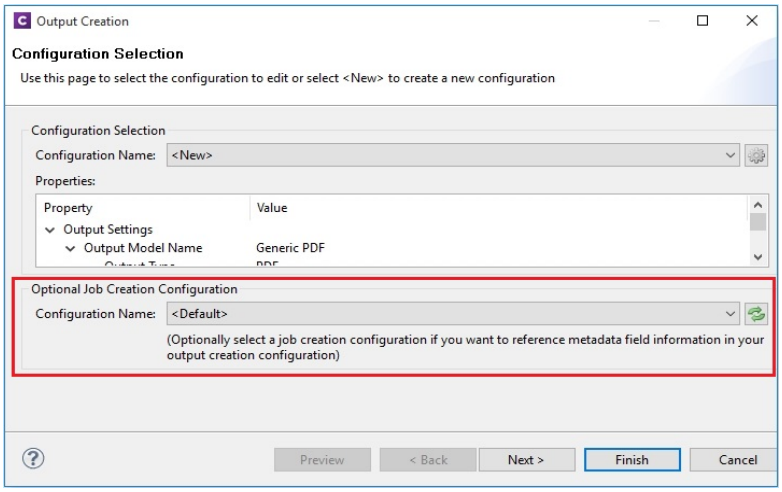
We have improved the way that output file names can be specified. A new dialog box has been added to the Print Wizard, to simplify the creation of Job Output Masks. While it is still possible to directly type a file name with placeholders in the Output File Mask box, it is now also possible to use the dialog to pick the metadata fields and other variables that can be used to create dynamic file names. (SHARED-12173)



A typical use case for using place holders in an Output File Mask is while generating PDF's for archiving purposes. This can require generating one PDF per document and often the files have to be named in a meaningful manner, by using an invoice number in the file name for instance. This requires one to define the invoice number as metadata in a Job Preset and then this metadata field can be used in the Output File Mask of the output preset. In addition, the Separation setting of the Output Preset has to be set to separate at the document level.

So the next time you need a dynamically generated output file name like `inv-${document.metadata.InvNumber}.pdf` or `${document.metadata.ID}-${system.time,'yyyy-MMM-dd'}.${template.ext}`, you can use this dialog to help you get what you need.

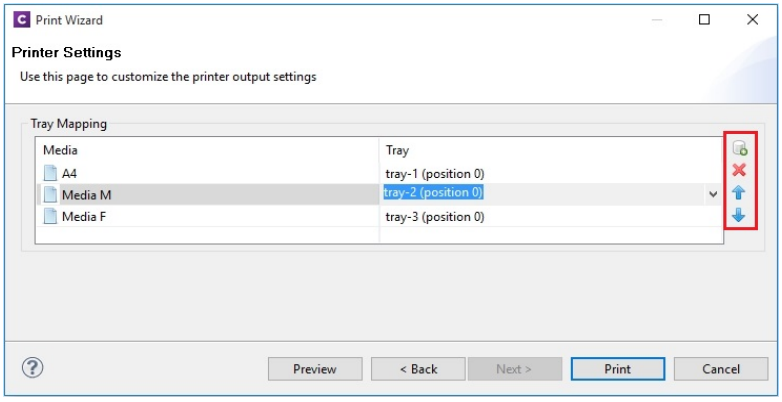
For the Output Preset to know what metadata is available, you can select a Job Preset when creating or modifying an Output Preset:



In the Advanced mode of the Print Wizard this new dialog works a bit different, because the metadata can be directly edited in the same wizard instead of having to refer to a Job Preset.

Tray Mapping for Multiple Templates

For printing to a cutsheet printer, the Output Preset allows mapping of media defined in a template to trays and media known by the printer. To make it easier to use an Output Preset for multiple templates, the list of media shown on the Tray Mapping page is no longer fixed. So now it is possible to easily define a tray mapping for all media used on a certain printer. (SHARED-49357)



This doesn't mean that all these media have to be used in every job, so one might even map multiple media types to the same tray. In such cases, a Job Preset could be used to filter jobs in

such a way that no conflicting tray mappings can occur within a job, as Job Presets allow filtering by media type.

Print Output

- **Improvements made to the Print Wizard**
  These include:
  - Improved usability in Inserter dialog. (SHARED-38279)
  - Data Filtering dialog usability improved. (SHARED-38281/38283)
  - Support added for manually setting both the horizontal and vertical gutter settings in Booklets (SHARED-53769)
  - The Additional Text and Barcode dialogs did not allow many of the available system fonts to be used. This issue has been fixed. (SHARED-46825)
- **Improved PDF comparison** implementation has improved output creation times. (SHARED-44097)
- **PostScript** Tray mapping configurations can now be made independent of the loaded template. (SHARED-49357)
- **IPDS and AFPDS** improvements:
  - Improved colour support. (SHARED-49554)
  - Extra image compression options added. (SHARED-51222)
- Improve output creation speed when **outputting with separation**. (SHARED-52088)
- **Soft masked images** were not handled correctly when writing to PDF/VT, causing errors. This issue has now been fixed. (SHARED-32335)

Known Issues

Installation Paths with Multi-Byte Characters

When installing the Traditional Chinese or Japanese versions of Connect, if the user specifies an alternative installation path containing multi-byte/wide-char characters it can break some of the links to the Connect-related shortcuts in the Start Menu and cause an error to appear at the end of the installer. The workaround for the moment is to use the default installation path. The problem will be addressed in a later release.

Switching Languages

Changing the language using the **Window>Preferences>Language Setting** menu option does not currently change all of the strings in the application to the selected language. This is a known issue and will be fixed in a later release.

In the meantime we offer the following workaround for anyone who needs to change the language:

1. Go to the .ini files for the Designer and Server Config:
    - C:\Program Files\Objectif Lune\OL Connect\Connect Designer\Designer.ini
    - C:\Program Files\Objectif Lune\OL Connect\Connect Server Configuration\ServerConfig.ini
2. Change the language parameter to the required one under Duser.language=en | es | de | fr | it | ja | pt | tw | zh

Only one of the above language tags should be selected. Once saved, Connect will appear in the selected language at next start-up.

GoDaddy Certificates

When installing Connect offline, dialogs allow installing the GoDaddy certificates. Most users should use the default settings and click **Next**. In some cases, however, this may not work correctly. For this reason those users should activate **Place all certificates in the following store** and then select the **Trusted Root Certification Authorities** as the target certificate store.

Available Printer Models

Note that only the single Printer Model (Generic PDF) will appear on the **Advanced** page of the **Print Wizard** by default.

To add additional printer models click on the settings button next to the Model selection entry box.

Colour Model in Stylesheets

The colour model of colours defined in a stylesheet can sometimes change after editing the stylesheet. This is a known issue and will be addressed in a subsequent release.

Online Help Links Point to Introductory Page

Context sensitivity for the online help is not yet enabled in Connect. All links and F1 calls point to the introductory page, where you can Search on keywords to bring up Help pages relating to the topic.

Context sensitivity will be introduced in a subsequent release of Connect.

Image Preview in Designer

If in the Windows Internet settings (**Connection Settings > LAN configuration**) a proxy is enabled, but "Bypass proxy settings for local addresses" is not checked, the image preview service, conversion service and live preview tab in the Designer will not work and exhibit the following issues:

- Images will be shown as 0 size boxes (no red 'X' is displayed)
- Live preview does not progress, and when re-activated reports "browsers is busy"

To fix the issue you must check the "Bypass proxy settings for local addresses" option.

VIPP Output

Some templates set up with landscape orientation are being produced as portrait in VIPP. It can also sometimes be the case that text and images can be slightly displaced. These are known issues and will be addressed in a later release of Connect.

PrintShop Mail Connect Release Notes 1.6.1

Overview

This document provides an overview of the new features and enhancements in PrintShop Mail Connect 1.6.1.

The major focus of Connect 1.6 has been to improve performance and increase stability, as well as launching a new option for the PReS Connect and PlanetPress Connect brands called **OL Connect Send**.

Installing PrintShop Mail Connect 1.6.1

- PrintShop Mail Connect is released as a 64 Bit version only.
- Full details on installing and licensing PrintShop Mail Connect can be found in the online help in the installer.
- Note that PrintShop Mail comes with a 30 day trial licenses by default.

Connect 1.6.1 General Enhancements and Fixes

Performance improvements

- Changes to the handling of **transparency in PDF backgrounds** has not only cured some job failures, but has also led to substantial improvements to both output speeds and filesizes. (49680)
- Improvements made to the **clean-up** processes, improving overall production speed.
- Some **memory leaks plugged**, improving overall production speed.
- Improved reliability when using **MS-SQL** as back-end database.

Further Performance improvements can be found detailed in the Output Enhancements and Fixes section.

Documentation improvements

- Help layout changed to allow **easier navigation**, and the content improved.
- Broken links within the **Welcome Screen** have been fixed. (39077/47964)

Installer improvements

- **Microsoft SQL Server** can now be setup as the back-end database during the installation process. (47546)

Print Manager improvements

- Support added for **IPDS printer status (SNMP)**. (51629)
- Improvements with **handling and deleting of overlays** for some IPDS printers. (48961/49508)
- Disable printer capabilities check when the printer is busy, to reduce connection problems with **older printers**. (51589)
- Improved support for **Xerox IPDS 495** printers. (48774)
- Improved error handling when **IPDS structural errors are encountered** in IPDS resources. The Print Manager now reports the error and ceases sending the job to the printer. (50138)

- Print Manager would hang when **unexpected errors encountered on IPDS printers**. The Print Manager now checks for fatal errors and exits the running job when one is encountered. (50135)
- Improved robustness when encountering **Communications Errors**. For example, the errors previously encountered when removing a job whilst Print Manager was retrying a connection. (49757)

Connect 1.6.1 Designer Enhancements and Fixes

General Designer improvements

- **Interface improvements** such as inclusion of icons for different types of files (js and CSS).
- Provided option to configure the **script timeout period**. (48639)
- Minor issues with non-English language **translations** fixed.
- Display issues that were sometimes encountered when changing **section background images** have been fixed.
- Issue with Virtual Stationery not being enabled when adding VS PDF in **Formal Letter Template** wizard now fixed. (47268)
- Fixed an issue in which **resizing an abs box** in some scenarios would result in an additional box being displayed. (47858)
- When **typing in a non-English input language** (such as Hebrew), the keyboard input would periodically switch to English. This issue has been fixed. (49566)
- Saving a template when in Source view would erroneously **show snippet content** within the Source view thereafter. This has been fixed. (50131)
- HEX encoding option added to **DataMatrix** barcodes. (47897)

Connect 1.6.1 Output Enhancements and Fixes

General

- Merge Engine **memory leaks** fixed. (50188)
- Improved creation of **Metadata** after content creation.
- Improved conditional content for **Email output**.
- Some templates containing JPG or Bitmap graphics could trigger the **flatten.exe** program (used for flattening external files) to repeatedly open, causing performance degradation or even failure. (51251)

Email Output

- Fixed issue with the **email Subject field** not being encoded properly when using characters other than Latin characters. (48781)
- To **improve privacy** certain Meta tags that were embedded in the output email HTML have been removed. These include "email-reply-to", "email-from", "sender-name" and "sender-address". (49864)
- The **Date field** was not always included in the email header. This has been fixed and the Date field should now be present in all email headers. (48706)

Print Output

- Improved processing speed for jobs involving **separation** that create multiple output files. (49167)
- Improvements made in the **conversion of PDFs** to other formats have made such conversions significantly faster and the output smaller, in PCL, AFP and IPDS output. (50140)
- Improved **page range handling** on last page.
- Improved **N-Up** positioning.
- **Improved color support** on older AFPDS and IPDS printers.
- **PDF and TIF objects** are now marked for optimization in AFPDS output
- **Vector graphic** processing improved, resulting in much faster output, and smaller file sizes. (50140)
- Issues with **AFPDS Highlight colors** have been resolved. (48397)

- Issues resolved with **missing non-Latin characters** (such as the dyet "Đ" character) in IPDS PDF Pass-through jobs. (51509)
- If an image file were replaced on disk between runs then a subsequent **Proof printing** would display the old image, . This has been fixed. (47567)
- Merge engine no longer slows down when using **external JavaScript** files in print sections. (48447)
- Fixed "**ApplicationException: Null**" errors encountered in some PCL outputs. (50868)
- Added support for **shaded bitmap fonts** to allow printing of grey text on supported IPDS and AFPDS printers. (49702)
- Fixed issues with output to IPDS Printers that do not support **FS45**. (48967)
- Fixed issues with imported PDF graphics that were **inverted in IPDS output**. (51504)
- Conversion of certain **FS10 G4 encoded images** in IPDS output was causing Xerox 495 Printers to crash. This printer specific issue has now been fixed. (51308)
- In IPDS and AFPDS output **lines and boxes** would be drawn using "Graphics Object Content Architecture" (GOCA) processing, rather than the more efficient Write Text (WT) processing. This has now been fixed, and the more efficient process is now used. (51127)
- New PrinterDef designed for **Nipson 7000** IPDS printer, which does not support Extended OVL. (50154)
- When converting a PDF in IPDS processing, if the **final item** could not be converted it was lost. This has now been fixed. (51325)

Known Issues

Installation Paths with Multi-Byte Characters

When installing the Traditional Chinese or Japanese versions of Connect, if the user specifies an alternative installation path containing multi-byte/wide-char characters it can break some of the links to the Connect-related shortcuts in the Start Menu and cause an error to appear at the end of the installer. The workaround for the moment is to use the default installation path. The problem will be addressed in a later release.

Switching Languages

Changing the language using the **Window>Preferences>Language Setting** menu option does not currently change all of the strings in the application to the selected language. This is a known issue and will be fixed in a later release.

In the meantime we offer the following workaround for anyone who needs to change the language:

1. Go to the .ini files for the Designer and Server Config:
    - C:\Program Files\Objectif Lune\OL Connect\Connect Designer\Designer.ini
    - C:\Program Files\Objectif Lune\OL Connect\Connect Server Configuration\ServerConfig.ini
2. Change the language parameter to the required one under Duser.language=en | es | de | fr | it | ja | pt | tw | zh

Only one of the above language tags should be selected. Once saved, Connect will appear in the selected language at next start-up.

GoDaddy Certificates

When installing Connect offline, dialogs allow installing the GoDaddy certificates. Most users should use the default settings and click **Next**. In some cases, however, this may not work correctly. For this reason those users should activate **Place all certificates in the following store** and then select the **Trusted Root Certification Authorities** as the target certificate store.

Available Printer Models

Note that only the single Printer Model (Generic PDF) will appear on the **Advanced** page of the **Print Wizard** by default.

To add additional printer models click on the settings button next to the Model selection entry box.

Colour Model in Stylesheets

The colour model of colours defined in a stylesheet can sometimes change after editing the stylesheet. This is a known issue and will be addressed in a subsequent release.

Online Help Links Point to Introductory Page

Context sensitivity for the online help is not yet enabled in Connect. All links and F1 calls point to the introductory page, where you can Search on keywords to bring up Help pages relating to the topic.

Context sensitivity will be introduced in a subsequent release of Connect.

Image Preview in Designer

If in the Windows Internet settings (**Connection Settings > LAN configuration**) a proxy is enabled, but "Bypass proxy settings for local addresses" is not checked, the image preview service, conversion service and live preview tab in the Designer will not work and exhibit the following issues:

- Images will be shows as 0 size boxes (no red 'X' is displayed)
- Live preview does not progress, and when re-activated reports "browsers is busy"

To fix the issue you must check the "Bypass proxy settings for local addresses" option.

VIPP Output

Some templates set up with landscape orientation are being produced as portrait in VIPP. It can also sometimes be the case that text and images can be slightly displaced. These are known issues and will be addressed in a later release of Connect.

PrintShop Mail Connect Release Notes 1.5

Overview

This document provides an overview of the new features and enhancements in PrintShop Mail Connect 1.5.

Installing PrintShop Mail Connect 1.5

- PrintShop Mail Connect is released as a 64 Bit version only.
- Full details on installing and licensing PrintShop Mail Connect can be found in the online help in the installer.
- Note that both PrintShop Mail come with 30 day trial licenses by default.

Connect 1.5 Designer Enhancements and Fixes

General Designer improvements

- A **color selection eyedropper** has been added, to allow the selection of a color from elsewhere on screen. (SHARED-33561/33646/36293)
- **Improved responsiveness** within the Designer, particularly when dealing with large and complex documents. (SHARED-44309)
- A **configurable Auto Save functionality** has been added for templates. (SHARED-40942/42085)
- Improvements made to **image file selection** functionality. (SHARED-42231/42451/42503/42556//43778)

Simplified creation of templates based on existing PDFs

- Option added to allow the creation of a **new print document based on an existing PDF**. (SHARED-19220)
- Improved support for **adding PDF files as Section backgrounds**. Files can now be referenced from disk or imported into Template. (SHARED-42496)
- Added support for drag and dropping **Data Fields** directly onto the page as absolutely positioned textboxes. (SHARED-43311)

Print Layout improvements

- **Page Number** formatting options (start/stop page numbering for sections, set numbering notation) improved in Print Section Properties dialog. (SHARED-39048)
- Added **repeating background images** support for print documents. (SHARED-43201)
- Option added to allow the insertion of **absolute positioned tables** on a master page. (SHARED-21967)

Email enhancements

- **User-definable SMTP settings**. New defaults are added for Sendgrid and Mailgun (in addition to Mandrill). (SHARED-43897)
- The standard **New email** wizard has been replaced with the new **Basic Email template** wizard. The new wizard has improved HTML structure. (SHARED-43338)
- Sending a **test email** no longer requires data. (SHARED-41889)
- **Tighter compression** for PDF attachments that are based on a print section. (SHARED-38575)
- **Colour picker** support added to the Email template wizards. (SHARED-33561)
- Added support for **PNG barcode** images in email messages. (SHARED-43787)

Barcode enhancements

- **Improved Barcode creation** with improved dialogs, better data validation and better error messaging. (SHARED-39295/42879)
- Font controls added to the **Barcode Properties** dialogs. (SHARED-22722/43659)
- Barcode improvements made in **Preview** mode. Support added for resizing and dragging of absolute positioned barcode objects, as well as resizing of inline barcode objects. (SHARED-43641)
- Barcodes can now have **transparent backgrounds**. (SHARED-43659)

Scripting improvements

- New ***closest()* command** added to the Scripting API, to locate closest matching element above it in the Document Object Model (DOM) tree. (SHARED-41789)
- **Script editing improved**. Line numbering now available within the editor, support for code completion and syntax highlighting added, as well as support for various ECMA6 commands. (SHARED-42768/43696)
- Support added for **cloning Sections in a Control Script** to allow a document to have a dynamic number of Sections. (SHARED-43683)
- Improved **Scripts tool tip** warning and error messaging. (SHARED-42550/43758) Improvements include:
    - Better tailored error messages and warnings.
    - Icons added representing script type as well as showing the issue severity.

    - Duplicated problems now filtered out.
    - Several other minor improvements.
- Improved support for raw HTML within Designer scripting API commands. (SHARED-43075)

Connect 1.5 Output Enhancements and Fixes

General

- Improved **content creation** processing speed for templates featuring PDF backgrounds. (SHARED-44350)

Email Output

- **Basic Email Action** wizard now made the default for new Email templates. (SHARED-43338)
- Support added for **user defined** SMTP/Email Service Provider (ESP) settings. (SHARED-43897)

Print Output

- New option added, allowing printing to **Windows Printer Driver**. (SHARED-35536)
- Improvements made to **external sort** option in Job presets. Support added for using input/output file placeholders. (SHARED-40944)
- New **HCF** file added that supports “top down wrap around sequence marks”. (SHARED-42326)
- Use **PostScript Media** name values in the PostScript DSC comments, to improve subsequent searches. (SHARED-42826)
- Option added to allow **storing of job resources** on PostScript printer’s own storage medium. (SHARED-43467)
- **OMR marks improved**, with support added for Match Numbers (Match Code, MC). (SHARED-43589)
- A **Proof preview function** has been added to the Output Wizard, to display onscreen how the current print job would appear when printed. (SHARED-43885)
- **Imposition improvement**. Can now set specific starting position via new Offset option. (SHARED-44022)
- Minor glitches in **Booklet** and **Imposition** output addressed. (SHARED-44340/44430)
- IPDS output: Added **support for levels and metadata**. Future versions of the Print Manager will be able to search and restart based upon these values. (SHARED-44173)

Connect 1.5 General Enhancements and Fixes

Installer improvements

- Improvements made to installation robustness. The installer now copes better when encountered **permissions issues** during installation. (SHARED-43732/43737)
- The **Update Client** has been updated to 1.1.9 and has been included in the Connect 1.5 installation. (SHARED-47065)

Print Manager improvements

- **Removing all finished jobs** for a specific printer with a single step. (SHARED-42984)
- **Retry counters** added to interface. (SHARED-44772)
- Minor dialog **updates and improvements** throughout the Print Manager. (SHARED-30040/41975/42189/42804/43890/44113/44214)

Known Issues

Installation Paths with Multi-Byte Characters

When installing the Traditional Chinese or Japanese versions of Connect, if the user specifies an alternative installation path containing multi-byte/wide-char characters it can break some of the links to the Connect-related shortcuts in the Start Menu and cause an error to appear at the end of the installer. The workaround for the moment is to use the default installation path. The problem will be addressed in a later release.

Switching Languages

Changing the language using the **Window>Preferences>Language Setting** menu option does not currently change all of the strings in the application to the selected language. This is a known issue and will be fixed in a later release.

In the meantime we offer the following workaround for anyone who needs to change the language:

1. Go to the .ini files for the Designer and Server Config:
    - C:\Program Files\Objectif Lune\OL Connect\Connect Designer\Designer.ini
    - C:\Program Files\Objectif Lune\OL Connect\Connect Server Configuration\ServerConfig.ini
2. Change the language parameter to the required one under Duser.language=en | es | de | fr | it | ja | pt | tw | zh

Only one of the above language tags should be selected. Once saved, Connect will appear in the selected language at next start-up.

GoDaddy Certificates

When installing Connect offline, dialogs allow installing the GoDaddy certificates. Most users should use the default settings and click **Next**. In some cases, however, this may not work correctly. For this reason those users should activate **Place all certificates in the following store** and then select the **Trusted Root Certification Authorities** as the target certificate store.

Available Printer Models

Note that only the single Printer Model (Generic PDF) will appear on the **Advanced** page of the **Print Wizard** by default.

To add additional printer models click on the settings button next to the Model selection entry box.

Colour Model in Stylesheets

The colour model of colours defined in a stylesheet can sometimes change after editing the stylesheet. This is a known issue and will be addressed in a subsequent release.

Online Help Links Point to Introductory Page

Context sensitivity for the online help is not yet enabled in Connect. All links and F1 calls point to the introductory page, where you can Search on keywords to bring up Help pages relating to the topic.

Context sensitivity will be introduced in a subsequent release of Connect.

Image Preview in Designer

If in the Windows Internet settings (**Connection Settings > LAN configuration**) a proxy is enabled, but "Bypass proxy settings for local addresses" is not checked, the image preview service, conversion service and live preview tab in the Designer will not work and exhibit the following issues:

- Images will be shows as 0 size boxes (no red 'X' is displayed)
- Live preview does not progress, and when re-activated reports "browsers is busy"

To fix the issue you must check the "Bypass proxy settings for local addresses" option.

VIPP Output

Some templates set up with landscape orientation are being produced as portrait in VIPP. It can also sometimes be the case that text and images can be slightly displaced. These are known issues and will be addressed in a later release of Connect.

PrintShop Mail Connect Release Notes 1.4.2

Overview

This document provides an overview of the new features and enhancements in PrintShop Mail Connect 1.4.n.

Installing PrintShop Mail Connect 1.4.n

- PrintShop Mail Connect is released as a 64 Bit version only.
- Full details on installing and licensing PrintShop Mail Connect can be found in the online help in the installer.
- Note that both PrintShop Mail come with 30 day trial licenses by default.

Connect 1.4.2 Enhancements and Fixes

Designer

- A blank page is no longer added to beginning of templates that use scripting to add pages from PDF files. This problem only appeared when saving to a new file from within Preview mode, or when generating output from Preview mode. (SHARED-44564)
- Image elements (`<img>`) referencing a PDF image would multiply when switching back and forth between Live and Preview modes, in email context. This has now been fixed. (SHARED-44066)

Email Output

- Elements whose style was set to `display: none` would be removed from HTML output, rather than just not being displayed. This error has been fixed. (SHARED-44151)

> **Note**
>
> Elements hidden via the Conditional Script wizard are removed from the output.

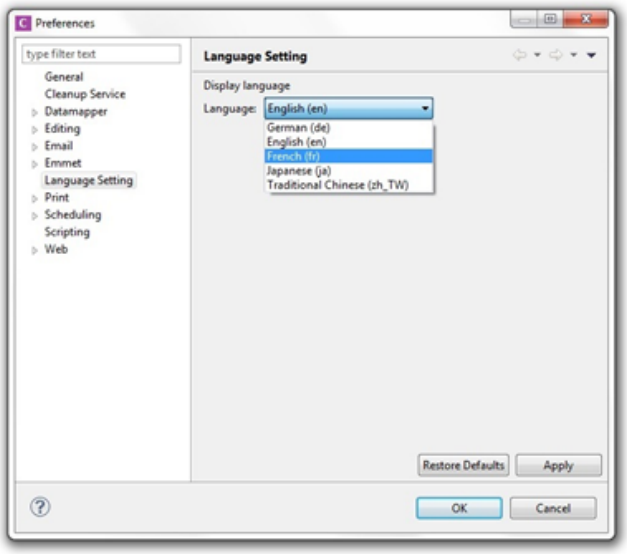
Print Manager

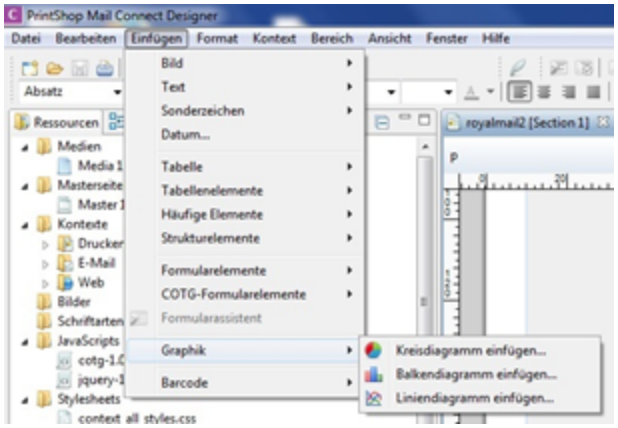
Connect 1.4.1 New Features and Enhancements

New Languages Added

The Connect user interface is now supported in Spanish, Italian, Portuguese and Chinese (Simplified) as well as English, French, German, Japanese and Chinese (Traditional). The default language remains English. Further languages will be introduced in later releases.

The language can be selected during the installation of Connect or via the Language Setting options in the Preferences dialog (note that Connect needs to be restarted in order to apply the selected language).



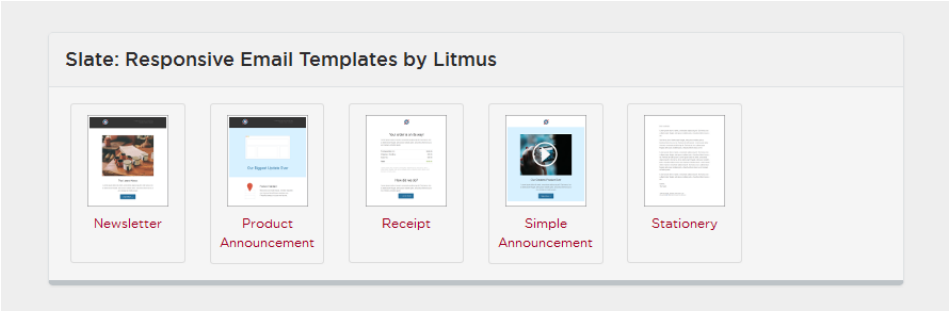


At present only the Connect user interface has been translated. Error messages and warnings will be translated for a later release.

Welcome Screen Extended

- The Printer Definition Configs and HCF files available on the OL Connect website are now grouped by manufacturer, to simplify selection.

- Connect 1.4.1 also introduces **Responsive Email Templates**.



Virtualisation

- Connect is now supported on the Microsoft Hyper-V and Hyper-V/Azure environments as well as the VMWare Workstation, Server, Player and ESX infrastructure environments.

Modifying Connect Installations

- Connect 1.4.1 introduces the ability to Modify Connect installations, once Connect 1.4.1 has been installed.

Connect 1.4.1 Designer Enhancements and Fixes

Email Context

- Email context sections can be **enabled or disabled based upon data value**. (SHARED-33656)
- Email **port number** can now be specified as part of the host name. (SHARED-38008)
- **New template wizard** for Slate templates by Litmus. (SHARED-36843)

Print Context

- Ability added to **mirror margins on back pages of Duplex jobs**, via Facing Pages selection added in Sheet Configuration dialog. (SHARED-40505)
- Can now suppress Master Page on duplex back pages, if there are no contents.

Scripting

- Result can now be written as a JSON string into an attribute or text (instead of the field value). Useful for web contexts where a front-end script can easily read the value.
- User-defined formatting masks can now be used when outputting dates and numerical values.
- Conditional script set to "Show if Condition is true" will hide the object by default.
- Page range can be specified via scripting when setting PDF as background. (SHARED-36998)
- Matched elements are highlighted when hovering over scripts. (SHARED-38293)

General

- Numerous improvements to Designer GUI and user-experience. These include:
  - A new "**Styles**" pane that displays the styles applied to the selected object.
  - **Image Selection** Dialog now inserts from resources, files or URL.
  - **Line number** option now available from any source view.
  - View menu now has entries to **switch between views** within the Workspace.
- Set any **PDF as background** for a Section. (SHARED-39880)
- Specify a **background image** and control its size, position and repeat mode. (SHARED-14522)
- Option to **generate JSON string from data model fields** to pass data record information to client side script. (SHARED-39337)
- **CSS Class name completion** suggests CSS classes based upon the current section. (SHARED-36870)
- **CSS Style inspector** allows full control over styles. (SHARED-22929)

Connect 1.4.1 Output Enhancements and Fixes

Print Output

- Images, barcodes, OMRs and text can now all be added to the page at time of output generation.
- Ability to **add Metadata** to any output type (previously only PDF and AFP), for use within output Presets.

- **Static strings** can now be added to Metadata in job Presets.
- The **Output Path** in the Output Preset can now be set dynamically.
- New Color setting for the **Add Text** option. (SHARED-40830)
- **Monochrome** and **Dithering** support added to PCL output. (SHARED-39937)
- **Dithering** and **B&W threshold** values now supported in the Print Wizard. (SHARED-39520)
- New option to control **Compact Font Format** settings within PostScript. (SHARED-39902)
- Support for Simple Network Management Protocol **(SNMP) queries** added to Print Manager. (SHARED-39213)

Email Output

- **SMTP port** can now be customized when sending email.

Known Issues

Installation Paths with Multi-Byte Characters

When installing the Traditional Chinese or Japanese versions of Connect, if the user specifies an alternative installation path containing multi-byte/wide-char characters it can break some of the links to the Connect-related shortcuts in the Start Menu and cause an error to appear at the end of the installer. The workaround for the moment is to use the default installation path. The problem will be addressed in a later release.

Switching Languages

Changing the language using the **Window>Preferences>Language Setting** menu option does not currently change all of the strings in the application to the selected language. This is a known issue and will be fixed in a later release.

In the meantime we offer the following workaround for anyone who needs to change the language:

1. Go to the .ini files for the Designer and Server Config:
    - C:\Program Files\Objectif Lune\OL Connect\Connect Designer\Designer.ini
    - C:\Program Files\Objectif Lune\OL Connect\Connect Server Configuration\ServerConfig.ini
2. Change the language parameter to the required one under Duser.language=en | fr | de | ja | zh

Only one of the above language tags should be selected. Once saved, Connect will appear in the selected language at next start-up.

GoDaddy Certificates

When installing Connect offline, dialogs allow installing the GoDaddy certificates. Most users should use the default settings and click **Next**. In some cases, however, this may not work correctly. For this reason those users should activate **Place all certificates in the following store** and then select the **Trusted Root Certification Authorities** as the target certificate store.

Available Printer Models

Note that only the single Printer Model (Generic PDF) will appear on the **Advanced** page of the **Print Wizard** by default.

To add additional printer models click on the settings button next to the Model selection entry box.

Colour Model in Stylesheets

The colour model of colours defined in a stylesheet can sometimes change after editing the stylesheet. This is a known issue and will be addressed in a subsequent release.

Online Help Links Point to Introductory Page

Context sensitivity for the online help is not yet enabled in Connect. All links and F1 calls point to the introductory page, where you can Search on keywords to bring up Help pages relating to the topic.

Context sensitivity will be introduced in a subsequent release of Connect.

Image Preview in Designer

If in the Windows Internet settings (**Connection Settings > LAN configuration**) a proxy is enabled, but "Bypass proxy settings for local addresses" is not checked, the image preview service, conversion service and live preview tab in the Designer will not work and exhibit the following issues:

- Images will be shows as 0 size boxes (no red 'X' is displayed)
- Live preview does not progress, and when re-activated reports "browsers is busy"

To fix the issue you must check the "Bypass proxy settings for local addresses" option.

VIPP Output

Some templates set up with landscape orientation are being produced as portrait in VIPP. It can also sometimes be the case that text and images can be slightly displaced. These are known issues and will be addressed in a later release of Connect.

Knowledge Base

You can find extra information in Connect Knowledge Bases which complement the information contained in this manual.

The PrintShop Mail Connect Knowledge Base can be found here.

Click to download the EULA as PDF

Legal Notices and Acknowledgements

Warning: PrintShop Mail Connect is protected by copyright law and international treaties. Unauthorized reproduction or distribution of this program, via any means, in part or in whole, may be prosecuted to the full extent of the law.

The license agreements for the associated open source third party components can be found in the following installation folder: *C:\Program Files\Objectif Lune\OL Connect\Legal Notices*

This application uses the following third party components:

- **Adobe PDF Library** which is either a registered trademark or trademark of Adobe Systems Incorporated in the United States and\or other countries.
- **Adobe XMP Core** Copyright © 1999 - 2010, Adobe Systems Incorporated. All rights reserved.
- **ASM Core** library which is distributed under a BSD 3-clause License
- **c3p0** which is licensed under the terms of the Lesser General Public License (LGPL) Version 2.1. The source code can be obtained from the following location: https://github.com/swaldman/c3p0
- **DD Plist** which is distributed under an MIT license (MIT License).
- **Eclipse Gemini Blueprint** which is distributed under the terms of the Apache Software License Version 2.0. This product includes sub-components with separate copyright notices and license terms.
- **Eclipse Nebula** This application also uses the following Eclipse Nebula components which are distributed under the terms of the **Eclipse Public License (EPL) v 2.0** :
  - Radiogroup Widgets
- **Eclipse Persistence Services Project (EclipseLink)**, Copyright © 2007, Eclipse Foundation, Inc. and its licensors. All rights reserved. This is distributed under the terms of the Eclipse Public License Version 1.0 and Eclipse Distribution License Version 1.0.
- **Fugue Icons** by Yusuke Kamiyamane which are distributed under the terms of the Creative Commons Attribution 3.0 License.
- **Gecko** which is distributed under the terms of the Mozilla Public License (MPL) Version 2.0. Information on obtaining Gecko can be found on the following page: https://wiki.mozilla.org/Gecko:Getting_Started

NOTE: This library has been modified for Connect. To obtain copies of the modified library please contact your local Objective Lune Support team.

- **Glassfish** This application also uses the following Glassfish components which are distributed under the terms of the **Eclipse Public License (EPL) v 2.0** :
  - HK2 API module
  - HK2 Implementation Utilities
  - Javax Inject
  - Jersey
  - Jersey Bean Validation
  - Jersey Container Jetty HTTP
  - Jersey Container Servlet
  - Jersey Container Servlet Core
  - OSGi Resource Locator
  - ServiceLocator Default Implementation

  Information on how to download the Glassfish project sources can be obtained from here: https://mvnrepository.com/search?q=org.glassfish
- **Glassfish Java Mail** which is licensed under the terms of the Common Development and Distribution License (CDDL) Version 1.0. Information on how to download the Glassfish source can be obtained from here: https://wikis.oracle.com/display/GlassFish/Java+EE+7+Maven+Coordinates
- **Google Core Protocol Buffers** library which is distributed under a BSD 3-clause License
- **Hamcrest Matchers** Copyright © 2000-2006, www.hamcrest.org. All rights reserved.
- **HyperSQL**, Copyright © 2001-2010, The HSQL Development Group. All rights reserved.
- **IcoMoon**. Connect uses unmodified icons from IcoMoon (https://icomoon.io/#icons-icomoon) which have been made available under the Creative Commons By 4.0 license (https://creativecommons.org/licenses/by/4.0).
- **ICU4J 4.4.2** Copyright © 1995-2013 International Business Machines Corporation and others. All rights reserved.
- **J2V8** which is distributed under the terms of the Eclipse Public License (EPL) Version 1.0. The source code for J2V8 can be obtained from the following location: https://github.com/eclipsesource/j2v8

- **Jacob Java Com Bridge** which is licensed under the terms of the GNU Lesser General Public License (LGPL) Version 2. The source code for this can be obtained from the following location: http://sourceforge.net/projects/jacob-project/files/jacob-project/
- **Java Advanced Imaging Image I/O Tools** which is distributed under a BSD 2-clause License (BSD-2-Clause license)
- **JavaSysMon** Copyright © 2009 ThoughtWorks, Inc. All rights reserved.
- **Java XmlHttpRequest** which is licensed under the terms of the GNU Lesser General Public License Version (LGPL) 2.1. The source code for this can be obtained from the following location: https://github.com/objectifluneCA/java-XmlHttpRequest
- **JavaX EJB API** which is distributed under the terms of the Common Development and Distribution License (CDDL) Version 1.0. The source code for this can be obtained from the following location: https://mvnrepository.com/artifact/javax.ejb/javax.ejb-api/3.2.2
- **JavaX interceptor API** which is distributed under the terms of the Common Development and Distribution License (CDDL) Version 1.0. The source code for this can be obtained from the following location: https://mvnrepository.com/artifact/javax.interceptor/javax.interceptor-api/1.2.2
- **JavaX Mail** which is distributed under the terms of the Common Development and Distribution License (CDDL) Version 1.1. The source code for this can be obtained from the following location: https://java.net/projects/javamail/downloads/directory/source
- **JavaX Management J2EE API** which is distributed under the terms of the Common Development and Distribution License (CDDL) Version 1.0. The source code for this can be obtained from the following location: https://mvnrepository.com/artifact/javax.management.j2ee/javax.management.j2ee-api/1.1.2
- **JavaX Resource API** which is distributed under the terms of the Common Development and Distribution License (CDDL) Version 1.0. The source code for this can be obtained from the following location: https://mvnrepository.com/artifact/javax.resource/javax.resource-api/1.7.1
- **JavaX Servlet API** which is distributed under the terms of the Common Development and Distribution License (CDDL) Version 1.0. The source code for this can be obtained from the following location: https://mvnrepository.com/artifact/javax.servlet/javax.servlet-api/3.1.0
- **JavaX Transaction API** which is distributed under the terms of the Common Development and Distribution License (CDDL) Version 1.0. The source code for this can be obtained from the following location: https://mvnrepository.com/artifact/javax.transaction/javax.transaction-api/1.3

- **JavaX WS RS API** which is distributed under the terms of the Eclipse Public License v 2.0 (EPL2). The source code for this can be obtained from the following location: https://mvnrepository.com/artifact/javax.ws.rs/javax.ws.rs-api/2.1.1
- **JBCrypt** library which is distributed under the ISC License (ISC)
- **Jersey** which is distributed under the terms of the Common Development and Distribution License (CDDL) Version 1.1. Information on how to obtain the source code can be found at the following location: http://repo1.maven.org/maven2/org/glassfish/jersey/jersey-bom
- **jersey-json-1.13** which is licensed under the terms of the Common Development and Distribution License (CDDL) Version 1.1. Information on how to obtain the source code can be found at the following location: http://mvnrepository.com/artifact/com.sun.jersey/jersey-json/1.13-b01
- **Jersey Multipart** which is distributed under the terms of the Common Development and Distribution License (CDDL) Version 1.1. Information on how to obtain the source code can be found at the following location: http://repo1.maven.org/maven2/org/glassfish/jersey/jersey-bom
- **JNA Version 3.5.1** which is distributed under the terms of the GNU Lesser General Public License Version (LGPL) 2.1. The source code for this can be obtained from the following location: https://github.com/twall/jna/releases
- **Logback** which is distributed under the terms of the Eclipse Public License (EPL) Version 1.0. The source code for Logback can be obtained from the following location: https://logback.qos.ch/download.html
- **MariaDB** which is distributed under the terms of the GNU General Public License Version 2. Information on how to obtain the source code can be found at the following location: https://mariadb.org/get-involved/getting-started-for-developers/get-code-build-test/
- **MariaDB Java Client** which is distributed under the terms of the GNU Lesser General Public License Version (LGPL) 2.1. The source code for this can be obtained from the following location: https://mvnrepository.com/artifact/org.mariadb.jdbc/mariadb-java-client
- **Mchange Commons Java** which is licensed under the terms of the Lesser General Public License (LGPL) Version 2.1. The source code can be obtained from the following location: https://mvnrepository.com/artifact/com.mchange/mchange-commons-java
- **Minimatch Java** which is distributed under an MIT license (MIT License).
- **Objectweb ASM**, Copyright © 2000-2011 INRIA, France Telecom. All rights reserved.
- **Oracle JDBC Driver** which is licensed by the Oracle Free Use Terms and Conditions (FUTC) licence.

- **org.eclipse.equinox.weaving.springweaver** which is distributed under the terms of the Eclipse Public License (EPL) Version 1.0. The source code can be obtained from the following location: https://github.com/osgi-forks/dynaresume
- **PDF Renderer** which is licensed under the terms of the Lesser General Public License (LGPL) Version 2.1. The source code can be obtained from the following location: https://mvnrepository.com/artifact/pdf-renderer/pdf-renderer
- **Polyfills** which is licensed under the Unlicense license. The source code can be obtained from the following location: https://github.com/inexorabletash/polyfill.
- **RapidJSON** which is distributed under an MIT license.
- **Relique CSV Driver** which is licensed under the terms of the Lesser General Public License (LGPL) Version 2.1. The source code can be obtained from the following location: https://sourceforge.net/p/csvjdbc/code/ci/csvjdbc-1.0.31/tree/
- **Rhino 1.7R4 and 1.7.7.1** which are licensed under the terms of the Mozilla Public License (MPL) Version 2.0. The source code for these can be obtained from the following location: https://developer.mozilla.org/en-US/docs/Mozilla/Projects/Rhino/Download_Rhino
- **Saxon** which is distributed under the terms of the Mozilla Public License (MPL) Version 2.0. The source code for this can be obtained from the following location: http://sourceforge.net/projects/saxon/files/Saxon-HE/9.6/
- **Servlet API** developed by Sun as part of the Glassfish project and licensed under the terms of the Common Development and Distribution License (CDDL) Version 1.0. Information on how to download the Glassfish source (as part of Java EE platform) can be obtained from here: https://wikis.oracle.com/display/GlassFish/Java+EE+7+Maven+Coordinates
- **Simple Logging Facade for Java (SLF4J)** Copyright © 2004-2017 QOS.ch. All rights reserved.
- **SpotBugs Annotations** which is licensed under the terms of the GNU Lesser General Public License (LGPL) Version 2.1. The source code for this can be obtained from the following location: https://github.com/spotbugs/spotbugs
- **Spring Framework** which is distributed under the terms of the Apache Software License Version 2.0. This product includes sub-components with separate copyright notices and license terms.
- **StAX Utilities** Copyright © 2007, StAX Utilities Project. All rights reserved.
- **Stax2 API** Copyright 2010-2018 FasterXML.com.

- **Tern** which is distributed under the terms of the Eclipse Public License (EPL) Version 1.0. The source code for tern can be obtained from the following location: https://github.com/angelozerr/tern.java
- **Web Services Description Language for Java** which is distributed under the terms of the Common Public License v 1.0. The source code for this can be obtained from the following location: http://wsdl4j.cvs.sourceforge.net/viewvc/wsdl4j/
- **XStream Core** library which is distributed under a BSD 3-clause License
- **XULRunner** which is distributed under the terms of the Mozilla Public License Version 2.0. The source code for this can be obtained from the following location: http://ftp.mozilla.org/pub/mozilla.org/xulrunner/releases/latest/source/
- **zziplib** which is licensed under the terms of the Mozilla Public License (MPL) Version 1.1. The source code for this can be obtained from the following location: http://sourceforge.net/projects/zziplib/files/zziplib13/
- **7-Zip SFX** which is licensed under the terms of the GNU Lesser General Public License (LGPL) Version 2.1. The source code for this can be obtained from the following location: https://github.com/chrislake/7zsfxmm

Portions of certain libraries included in this application which are distributed under the terms of the **Mozilla Public License** have been modified. To obtain copies of the modified libraries please contact your local Objective Lune Support team.

Apache Software License Components

This application also uses the following components which are distributed under the terms of the **Apache Software License Version 2.0**:

- Apache ActiveMQ
- Apache Batik
- Apache Commons Beanutils
- Apache Commons CLI
- Apache Commons Codec
- Apache Commons Collections
- Apache Commons Compress
- Apache Commons DBCP
- Apache Commons Digester
- Apache Commons Imaging

- Apache Commons IO
- Apache Commons JCS Core
- Apache Commons Lang
- Apache Commons Logging
- Apache Commons Math
- Apache Commons Pool
- Apache Commons Text
- Apache Commons Validator
- Apache Derby
- Apache Felix and dependencies
- Apache Geronimo
- Apache HttpClient
- Apache HttpClient Mime
- Apache HttpCore
- Apache POI
- Apache ServiceMix
- Apache Tika Core
- Apache Tika Standard Parser
- Apache Xerces2 Java Parser
- Apache XML Graphics
- Apache XML Beans
- ASN1bean
- Barcode4j
- Google Collections
- Google GSON
- Hibernate Validator
- Jackcess
- Jackson JSON processor
- JCIP Annotations
- JDBI
- Jetty

- Jetty Websocket API
- JSON Path
- JSON Sanitizer
- JSON Small and Fast Parser
- Liquibase
- LMAX Disruptor
- Minidev
- Nimbus JOSE+JWT
- Objenesis
- OpenCSV
- OPS4J Pax Web
- org.eclipse.persistence.logging.slf4j
- org.json.simple
- PAC4J
- Quartz Scheduler
- Sisyphsu DateParser
- Sisyphsu Retree
- Snakeyaml
- SNMP4J
- Spring Dynamic Modules
- Tika
- UCanAccess
- Woodstox

Eclipse Technology:

This Software includes unmodified Eclipse re-distributables, which are available at www.eclipse.org. The Eclipse re-distributables are distributed under the terms of the Eclipse Public License - v 1.0 that can be found at https://www.eclipse.org/legal/epl-v10.html.

Eclipse Adoptium OpenJDK

This application uses unmodified re-distributables from the Eclipse Adoptium OpenJDK project. The Licenses used in that project are:

- Build scripts and other code to produce the binaries, the website and other build infrastructure are licensed under Apache License, Version 2.0.
- OpenJDK code itself is licensed under GPL v2 with Classpath Exception (GPLv2+CE).
- Eclipse OpenJ9 is licensed under Several licenses.

Further Components:

- Portions of this software are copyright © 2018 **The FreeType Project** (www.freetype.org). All rights reserved.
- This product includes software developed by **JSON.org** (https://www.json.org), which is distributed under the JSON License (JSON License)
- This product includes software developed by the Indiana University Extreme! Lab. For further information please visit http://www.extreme.indiana.edu/

Copyright Information

Copyright © 1994-2023 Objectif Lune Inc. All Rights Reserved.

No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any other language or computer language in whole or in part, in any form or by any means, whether it be electronic, mechanical, magnetic, optical, manual or otherwise, without prior written consent of Objectif Lune Inc.

Objectif Lune Inc.disclaims all warranties as to this software, whether expressed or implied, including without limitation any implied warranties of merchantability, fitness for a particular purpose, functionality, data integrity or protection.

PlanetPress, PReS and PrintShop Mail are registered trademarks of Objectif Lune Inc.