PReS®

Connect

OL™ Software

User Guide

Version: 1.4.2

PReS®
Connect
OL™ Software

User Guide
Version 1.4.2
Last Revision: 2016-06-30

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

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

Table of Contents

Welcome to PReS Connect 1.4.2

Note

In our continuous effort to facilitate your PReS Connect experience, our entire documentation is being overhauled. Since we are still in the process of restructuring all this information, expect to see our online help evolve frequently over the next few weeks. We apologize for any inconvenience, but know we are working hard to improve your experience as quickly as possible.

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

PReS Connect is a series of four tools designed to optimize and automate customer communications management. They work together to improve the creation, distribution, interaction and maintenance of your communications.

The PReS Connect **Datamapper** and **Designer** is designed to create output for print, email and the web within a single template and from any data type, including formatted print streams. Output presets applied outside the design phase make 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.

PReS Connect also includes a process automation server, called **Workflow**. It is capable of servicing response form web pages and email to provide interactive business communications.

PReS Connect can create documents for tablets and mobile devices that run a free **CaptureOnTheGo** App. Users with a CaptureOnTheGo subscription can then download documents to their own devices, interact with them and send the captured data back to PReS for conversion into additional documents or workflows.

This online documentation covers **PReS Connect** version 1.4.2.

Icons used in this guide

Icons are used throughout this guide to point your attention to certain information.

Note

Complementary information that is not critical, but may help you better use PReS Connect.

Tip

Information that is useful or suggests an easier method.

Technical

Information that may require specific knowledge to understand.

Warning

Information that is potentially critical to using PReS Connect. Pay close attention.

Setup And Configuration

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

- "System and Hardware Considerations" below
- "Installation and Activation" on page 17
- "Server Settings" on page 33

System and Hardware Considerations

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

- "System Requirements" below
- "Environment considerations" on the next page
- "Database Considerations" on page 13
- "Network considerations" on page 13
- "Language and Encoding considerations" on page 14
- "Performance Considerations" on page 14

System Requirements

These are the system requirements for PReS Connect 1.4.2

Operating System (64-bit only)

- Microsoft Windows 2008/2008 R2 Server
- Microsoft Windows 2012/2012 R2 Server
- Microsoft Windows Vista
- Microsoft Windows 7
- Microsoft Windows 8.1
- Microsoft Windows 10

> **Note**
>
> Windows XP, Windows 2003 and older versions of Windows are not supported by PReS

Connect.

Minimum Hardware Requirements

- NTFS Filesystem (FAT32 is not supported)
- CPU Intel Core i7-5960X @ 3.00GHz (8 core with Hyper-Threading)
- 16GB RAM
- Disk Space: At least 10GB (20GB recommended)

Note

For tips and tricks on performance, see "Performance Considerations" on page 14.

Environment considerations

Virtual Machine Support

PReS Connect supports VMWare Workstation, VMWare Server, VMWare Player, VMWare ESX, Microsoft Hyper-V and Microsoft Hyper-V/Azure infrastructure environments as software installed on the Guest operating system. PReS Connect does not officially support VMotion under ESX at the moment.

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 VMWare (from VMWare, Inc.) and Microsoft virtual machine environments are supported, other virtual environments (such as Parallels, Xen and others) are not supported at this time.

Terminal Server/Service

PReS 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 PReS Connect is located, unexpected behaviours may occur and will not be supported by Objectif Lune Inc.. Furthermore, using PReS Connect in a Terminal Service environment is an infringement of our End-User License Agreement.

Remote Desktop

Tests have demonstrated that PReS 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.

PReS Connect 1.3 and later have been certified under Remote Desktop.

64-bit Operating Systems

PReS 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 PReS Connect. Please see KB-002: Antivirus Exclusions 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 completely, you may re-enable the service and start it.

Database Considerations

This page describes the different considerations and pre-requisites for the database back-end used by PReS Connect, whether using the MySQL instance provided by the installer, or pre-existing instance.

Using the MySQL Instance from the Installer

The MySQL Instance provided in the Installation Wizard is already pre-configured with options to provide the most stable back-end setup.

These are the specific options that have been changed in our version of "*my.ini*":

- **max_connections = 200** : PReS 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 PReS 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.

Using a pre-existing MySQL Instance

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

- The MySQL account must have access to all permissions using the GRANT Command, including creating databases.
- The database configuration must include the options detailed in the "Using the MySQL Instance from the Installer" above section.
- 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.

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.

Language and Encoding considerations

Please note the following considerations:

- **Language**:
  - PReS 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
    - Portuguese
    - Chinese (Simplified)
    - Chinese (Traditional)
    - Japanese.

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

Performance Considerations

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

Performance Analysis Details

In order to get the most out of PReS Connect, it is important to determine how best to maximize performance. The following guidelines will be helpful in extracting the best performance from

PReS Connect before looking into hardware upgrades or extra PReS Connect performance packs.

- **Job Sizes and Speed**: In terms of pure output speed, it's important to first determine what job size is expected, and adjust Scheduling Preferences accordingly. The basic rules are:
    - If processing a small number of very large records (when each individual record is composed of a large number of pages), more instances with an equal amount of speed units is better. For hardware, RAM and Hard Drive speeds are most important, since the smallest divisible part (the record) cannot be split on multiple machines or even cores.
    - If creating a very large number of small records (hundreds of thousands of 2-3 page individual records, for instance), a smaller number of instances with a large number of speed packs would be better. As for hardware, then the number of cores becomes critical, whereas RAM and hard drive are secondary. Performance Packs, as well as the MySQL instance being separate, would be helpful if your most powerful machine starts struggling.
    - Mix and match. For example, one instance prioritized for large jobs and the rest for smaller, quicker jobs. Or the contrary. Or, whatever you want, really.
- **RAM Configuration**: By default, each instance of the Merge Engine and Weaver Engine is set to use 640MB of RAM. This means that regardless of speed units, if not enough memory is available, output speed might not be as expected. Assuming that the machine itself is not running any other software, the rule of thumb is the following: The total number of used memory in the machine should be pretty much the maximum available (around 95%).
  For each engine, it's necessary to modify the .ini file that controls its JAVA arguments. Edit as follows:
    - For the Merge Engine: see C:\Program Files\Objectif Lune\OL Connect\MergeEngine\Mergeengine.ini
    - For the Weaver Engine: see C:\Program Files\Objectif Lune\OL Connect\weaverengine\Weaverengine.ini
    - The parameters are -Xms640m for the minimum RAM size, -Xmx640m for the maximum RAM size. Explaining Java arguments is beyond the scope of this document. Please read references here, here and here for more details (fair warning: these can get pretty technical!).
- **Template and data mapping optimization**: Some functionality offered by the Designer modules are very useful, and sometimes downright awesome, but can cause the generation of records and of contents items to slow down due to their nature. Here are some of them:

- **Preprocessor and Postprocessor scripts**: manipulating data using a script may cause delays before and after the data mapping action has actually taken place, especially file conversion and data enrichment from other sources.
- **Loading external and network resources**: In Designer, 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 100,000 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 27 hours.
- **External JavaScript Librairies**: While loading a single javascript library from the web is generally very fast (and only done once for the record set), actually running a script on each generated page can take some time. Because yes, JavaScript will run for each record, and often take the same time for each record.
- **Inefficient Selectors**: Using very precise ID selectors in script wizards can be much faster than using a text selector, especially on very large documents. (more details on this in another upcoming page).
- **Complex Scripts**: Custom scripts with large, complex or non-optimized loops can lead to slowing down content creation. While it is sometimes difficult to troubleshoot, there are many resources online to help learn about JavaScript performance and coding mistakes. Here, here, and here are a few. 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.

High-Performance Hardware

The following is suggested when processing speed is important. Before looking into a Performance Packs to enhance performance, ensure that the below requirements are met.

- **A physical, non-virtualized server**. ESX and VMWare servers are great for reducing the numbers of physical machines in your IT space, but they must share the hardware between each other. While you can create a virtual machine that seems as powerful as a physical, it will still be sharing hardware with any other virtual machines, and this will adversely affect performance.
- **MySQL Database on a separate machine**. MySQL's main possible bottleneck is file I/O, and as such a high-performance setup will require this server to be on a separate machine, ideally with a high-performance, low-latency hard drive. A Solid State Drive (SSD) would be recommended.
- **High-Quality 16+ GB Ram.** This is especially true when working with many server instances ("speed units") running in parallel. The more parallel processing, the more RAM is recommended.

- **4 or 8 physical cores.** We're not talking Hyper-Threading here, but physical cores. Hyper-Threading is great with small applications, but the overhead of "switching" between the virtual cores, and the fact that, well, they're virtual, means the performance is much lesser on high-power applications such as OL Connect. In short, a dual-core processor with Hyper-Threading enabled is not equivalent to a quad-core processor.

Installation and Activation

This section provides detailed information about the installation and activation of PReS Connect 1.4.2.

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

PReS Connect 1.4.2 is comprised of 2 different installers: one for the PReS Connect software and one for PReS Workflow 8.

Where to Obtain the Installers:

The installers for PReS Connect 1.4.2 and PReS Workflow 8 can be obtained on DVD or downloaded as follows:

- If you are a **Customer**, the installers can be downloaded from the Objectif Lune Web Activations page: http://www.objectiflune.com/activations
- If you are a **Reseller**, the installers can be downloaded from the Objectif Lune Partner Portal: http://extranet.objectiflune.com/

For information on licensing, please see Activating your license.

Installation Pre-Requisites

- Make sure your system meets the System requirements.
- PReS Version 1.4.2 can be installed under a regular user account with Administrator privileges.
- You **must** install on an NTFS file system.

- PReS requires Microsoft .NET Framework 3.5 already be installed on the target system.
- In order to use the automation feature in Version 1.4.2, you need to install PReS Workflow 8. This can be installed on the same machine as an existing PlanetPress® Suite 7.6 installation or on a new computer. For more information, please see Information about PReS Workflow 8.
- As with any JAVA application, the more RAM available, the faster the product will execute.

Users of Connect 1.1

In order for users of PReS Connect 1.1 to upgrade to any later version through the Update Manager it is necessary to install a later version (1.1.8 or later) of the Objectif Lune Update Client.

If you do not have such a version installed already, the next time you run your Update Client it will show that there is an update available of itself to Version 1.1.8 (or later).

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 is open. It will automatically update itself.

Once you have done this, PReS Connect 1.4.2 will become available for download.

> **Note**
>
> From PReS Connect Version 1.2 onwards, the new version (1.1.8) of the Update Client is included by default with all setups.

Installing PReS Connect on Machines without Internet Access

In order to install PReS 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 you must first ensure that all Windows updates have been installed on the host machine. You then need to 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 PReS Connect installer to the offline machine and start the installation as normal

Installation Wizard

Starting the PReS Connect installer

The PReS Connect installer may be supplied as an ISO image or on a DVD:

- If an ISO image, either burn the ISO onto a DVD or unzip the contents to a folder (keeping the folder structure)
- If on a DVD, either insert the DVD and initiate the installation from there or copy the contents to a folder (keeping the folder structure)

Navigate to the PReS_Connect_Setup_x64.exe or and double-click on it. After a short while the Setup Wizard will appear as a guide through the installation steps.

> **Note**
>
> PReS Connect requires prior installation of Microsoft .NET Framework 3.5 (please refer to https://www.microsoft.com/en-us/download/details.aspx?id=21 for more details).

Selecting the required components

After clicking the Next button, the component selection page appears, where the different components of PReS Connect can be selected for installation. Currently, the following are available:

- **PReS Connect Designer**: The Designer module. It may be used as a standalone with no other installed modules, but it will not have certain capabilities such as automation and commingling.
- **PReS Connect Print Manager**: The Print Manager module required for handling IPDS print output, as also available for managing PCL and PostScript output.

- **PReS Connect Server**: The Server back-end giving capabilities such as automation, commingling, picking. It saves all entities generated from the Automation module into a database for future use.
- **PReS Connect Server Extension**: A slave server for a PReS Connect Server module. When an extension is installed, it communicates with the master server and shares tasks. Only one Server or Server Extension can be installed per machine, not both.
- **MySQL Product**: The database used by the PReS Connect Engine to save its items. This item is optional if an existing MySQL server, either on the same computer or a separate server, is to be used.
- **Installation Path**: This is the location where modules are to be installed.

> **Technical**
>
> All instances of **PReS Connect Server** and **PReS Connect Server Extension** must use the same instances of MySQL. Although it is possible for each separate server installation to have its own MySQL instance, doing so will prevent the servers from functioning together in a "Server Clustering" on page 34 setup. However, this instance can be on any server whether it is a Server (master) or Server Extension (slave) machine.

The installer can also calculate how much disk space is required for installing the selected components as well as how much space is available:

- **Disk space required**: Displays the amount of space required on the disk by the selected components.
- **Disk space available on drive**: Displays the amount of space available for installation on the drive currently in the Installation Path.
- **Recalculate disk space**: Click to re-check available disk space. This is useful if space has been made available for the installation while the installer was open.
- **Source repository location**: Displays the path where the installation files are located. This can be a local drive, installation media, or a network path.

Selection Confirmation

The next page confirms the installation selections made. Click **Next** to start the installation itself.

End User License Agreement

The next page displays the End User License Agreement, which needs to be read and accepted before clicking **Next**.

MySQL Configuration

The **Default Database Configuration** page only appears if the *MySQL Product* module was selected in the *Product Selection* screen. It defines the administrative password for the MySQL server as well as which port it uses for communication. Note that the installer will automatically configure the *Server* to use the supplied password and port.

- **MySQL user 'root' Password**: Enter the password for the 'root', or administration account, for the MySQL server. The password must be at least 8 characters long and contain at least one of each of the following:
  - 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: "Th1sIs@K"

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

- **Confirm 'root' Password**: Re-enter to confirm the password. Both passwords must match for installation to continue.
- **TCP/IP Port Number**: The port on which MySQL will expect, and respond to, requests. 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 installation), the number is highlighted in red and a warning message is displayed at the top of the dialog.
- **Allow MySQL Server to accept non-local TCP connections**: Click to enable external access to the MySQL server. This is required if setting up clustering, or if MySQL Server

will need to be accessed from any other machine. It is also required if MySQL is on a separate machine than PReS Connect.

> **Note**
>
> The last option may represent a security risk if the machine is open to the internet. It is recommended to ensure your firewall blocks access to port 3306 from external requests!

The **Database Connection** page appears if the MySQL Product module was not selected. It defines the necessary information required to connect to an existing database.

- **Database Configuration**: Select the database type to use for the PReS Connect Engine. Currently, only MySQL is supported.
- **Administrator Username**: Enter the username for a user with administrative rights on the database. Administrative rights are required since tables need to be created in the database.
  If accessing a database on a different machine, the server must also be able to accept non-local TCP connections, and the username must also be configured to accept remote connection. For example, the "root" MySQL user entered as root@localhost is not allowed to connect from any other machine than the one where MySQL is installed.
- **Administrator Password**: Enter the password for the above user. The appropriate MySQL password **must** be entered or the Connect installation will fail.
- **TCP/IP Port Number**: Enter the port on which the database server expects connections. For MySQL, this is **3306** by default.
- **Database Host Name**: Enter the existing database server's IP or host name.
- **Server Schema/Table**: Enter the name of the MySQL database into which the tables will be created. The Connect standard name is "objectiflune".
- **Test Connection** button: Click to verify that the information provide into previous fields is valid by connecting to the database.

PReS Connect Server/Server Extension Configuration

The **Server Configuration** page is where the *Server* or*Server Extension* component is configured.

- **Run Server as**: Defines the machine username and password that the**PReS Connect Server** module's service uses. This account should be administrator on the local machine. Note that the "Server Security Settings" on page 33 dialog can only be executed from the user specified here.

-
    - **Username**: The username the service uses to login. If the machine is on a domain, use the format domain\username.
    - **Password**: The password associated with the username.
    - **Validate user** button: Click to verify that the entered username and password combination is correct and that the service is able to login.
      This button *must* be clicked and the user validated before the **Next** button becomes available.

The **Server Extension Configuration** page only appears when installing the PReS Connect Server Extension module. It defines the settings to be used to connect to the Master server (PReS Connect Server module):

- **Username**: Enter the username expected by the OL Connect Server.
- **Username**: Enter the username expected by the PReS Connect Server.
- **Password**: Enter the password associated with the selected username
- **Hostname: Enter the machine name or IP Address where the PReS Connect Server resides.**
- **Port**: Enter the port to use to communicate with the Connect Server. *Default: 9340*
- **Use Server authentication** checkbox: Check this option and enter appropriate User details, if the Server expects authentication.
- **Validate Connection** button: Click to verify that the entered username and password is correct and the service is able to login. This button *must* be clicked and the user validated for the **Next** button to be available.

Click **Next** to start the actual installation process. This process can take several minutes.

Completing the installation

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

- **Configure Update Check** checkbox: This option is enabled by default. It causes the **Product Update Manager** to run after the installation is complete. This allows configuring PReS 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.
- **Show Log...** : If an issue was encountered during the installation, click this button to obtain details. This information can then be provided to Objectif Lune for troubleshooting.

- 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, the following message will be displayed after clicking “Finish” in the setup:

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 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 Client” 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 your license for more information.

> **Technical**
>
> 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.

How to Run Connect Installer in Silent Mode

The trigger for the Connect Installer to run in silent mode is a text file with the fixed name install.properties, which is located either in the same folder as the PReS_Connect_Setup_x86_64.exe or in the unpacked folder of the installer.exe.

The file needs to be a properties file with the following line types:

- Comment Lines, starting with # (e.g. # The options to configure an external database)
- Key=Value pairs (e.g. install.product.0 = Connect Designer)

For supported keys, please refer to the next paragraph.

Required and Optional Properties

Here is an example of an install.properties file.

```
# Verbose logging

logging.verbose = true

# Product selection

install.product.0 = Connect Designer

install.product.1 = Connect Server

# Server settings

server.runas.username = Localadmin

server.runas.password = admin

# Database configuration

database.type = mysql

database.host = 192.168.116.10

database.port = 3308

database.username = root

database.password = admin

database.schema = my_ol
```

Verbose Logging (Optional)

By default, the **Silent Installer** will log in the same way as the GUI installer. That means logging of error and warnings, and certain information during database configuration. A more verbose logging can be switched on by using logging.verbose = true.

Product Selection (Optional)

By default, the Silent Installer will install all products which are visible to the user in the respective brand (except for the Server Extension, because only Server or Server Extension can be installed at the same time).

However, it is possible to define the products to be installed using their visible product names, and using a counter for the install.product property, e.g.

- install.product.0 = Connect Designer
- install.product.1 = Connect Server

Other possible product names are Connect Print Manager, Connect Server Extension and MySQL Product.

Server / Extension configuration (required if Server / Extension is selected for install)

For both, Server and Server Extension, the user credentials who will be running the Server service need to be provided:

- server.runas.username
- server.runas.password

Additionally for the Server Extension, some properties to define the Master Server are required:

- server.master.host
- server.master.port
- server.master.authenticate = true_or_false
- server.master.username
- server.master.password

Database configuration

If the MySQL Product is part of the installation, the following properties should be defined:

- database.type = mysql (required)
- database.password (required, needs to match the security rules)
- database.port (optional, the default is 3306. The defined port needs to be available.)

Additional properties are required when the configuration of an external MySQL database is required (this is the case if a Server product but not MySQL product is installed):

- database.host
- database.username

Optionally, the "schema" name can be defined (the default is objectiflune):

database.schema

Information about PReS Workflow 8

If you wish to use PReS Workflow (automation) in conjunction with PReS Connect, you will need to install PReS Workflow 8 onto the same machine. Workflow 8 is provided through a separate installer which is available on CD or for download as follows:

- If you are a **Customer**, the installer can be downloaded from the Objectif Lune Web Activations page: http://www.objectiflune.com/activations
- If you are a **Reseller**, the installer can be downloaded from the Objectif Lune Partner Portal: http://extranet.objectiflune.com/

PReS Workflow 8 can be installed in parallel on the same machine as an existing PlanetPress® Suite 7.x installation. Note however:

- If both versions need to be hosted on the same machine, PReS Workflow 8 should always be installed after the legacy PlanetPress® Suite 7.x installation.
- When uninstalling PReS Workflow 8, you may be prompted to repair your legacy PlanetPress® Suite 7.x installation.
- If PReS Workflow 8 has been installed alongside PlanetPress® Suite 7, Capture can no longer be used with Workflow 7. The plugins are now registered uniquely to Workflow 8 and the messenger for Workflow 7 is taken offline. It is only then possible to use Capture from PReS Workflow 8.
- PReS Workflow 8 and PlanetPress® Workflow 7 cannot run simultaneously, since only one version of the Messenger service can run at a time. In fact, no 2 versions of PlanetPress Workflow can on the same machine simultaneously, whatever version is installed.
- It is possible to switch between different versions running by shutting down one version's services and then starting the other. However, this is not recommended. There are no technical limitations that prevent processes from previous PlanetPress Workflow versions (as far back as Version 4) to run on PReS Workflow 8, removing the need to run both versions.

For more information on the licensing of Workflow 8, please see Activating your license.

Activating a License

PReS Connect and PReS Workflow 8 includes separate 30 day trial periods during which it is not necessary to have a license for reviewing basic functionality. If a modification to the license if required, such as to allow an extension to the trial period, or for extra functionality or plugins (e.g., the PReS Plugin for Workflow 8), then a new activation code will need to be requested.

Obtaining the PReS 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 should have a different Magic Number, thus require a separate license file to be functional.

To get the PReS Connect Magic Number, open the PReS Connect Designer application:

- Open the **Start Menu**
- Click on **All Programs**, then **Objectif Lune**, then **PReS Connect**
- Open the **PReS Connect Designer [version]** shortcut.
- When the application opens, if it has never been activated or the activation has expired, the **Software Activation** dialog appears:
  - **License Information section**:
    - **Magic Number**: Displays the PReS Connect Magic Number.
    - **Copy to Clipboard**: Click to copy the Magic Number to the clipboard. It can then be pasted in the activation request email using the CTRL+V keyboard shortcut.
  - **Licensed Products section**:
    - **Name**: Displays the name of the application or module relevant to this activation.
    - **Serial Number**: Displays 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)
    - **Web Activations**: Click to be taken to the online activation page (not yet functional).
  - **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 .olconnectlicense file, once it has been received.
  - **Install License**: Click to install the license and activate the software (only available when a license file is loaded).
  - **Close**: Click to cancel this dialog. If a license file has been loaded, it will not automatically be installed.

The **Software Activation** dialog can also be reached through a shortcut located in **All Programs**, then **Objectif Lune**, then **PReS Connect** and is named **Software Activation**. Since it does not load the software, it is faster to access for the initial activation.

Requesting a license

After getting the Magic Number, a license request must be done for bothPReS Connect and Workflow 8:

- **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 PReS Connect license file via email.
- **Resellers**can create an evaluation license via the 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 PReS 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 PReS Workflow 8

PReS Workflow 8 uses the same licensing scheme as PReS Connect. There are two ways of activating the license for Workflow 8 after saving it to a suitable location:

- If only PReS Workflow 8 is installed, double-click on the license for the PReS Workflow 8 License Activation dialog to open. Applying the license here activates all of the Workflow 8 components.
- If you have both PReS Workflow 8 and PReS Connect installed, it will not be possible to double-click on the license file as this will always open the PReS Connect Activations Tool. Instead, open PReS Workflow 8 manually and apply the license through the activations dialog within.

Activating PReS Connect

To activate PReS 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 start menu:

- Open the **Start Menu**
- Click on **All Programs**, then **Objectif Lune**, then **PReS Connect**
- Open the **PReS Connect Designer [version]** shortcut. The “PReS Connect Software Activation” tool displays information about the license and the End-User License Agreement (EULA).
- Click the **Load License File** button.
- Read the EULA and click 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.

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

Migrating to a new computer

Currently there are no special migration tools to move data from one PReS Connect installation to another.
Instructions for migration will be available for later versions of the software when those tools become available.

User accounts and security

Permissions for PReS Connect Designer

PReS Connect Designer does not require any special permissions to run besides a regular program. It does not require administrative rights and only needs permission to read/write in any folder where Templates or Data Mapping Configurations are located.

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

Permissions for PReS Connect Server

The PReS Connect Server module, used by the *Automation* module, requires some special permissions to run. These permissions are set during installation, in the *Engine Configuration* section of the Installation Wizard, but it can also be configured later by modifying permissions for the service. To do this:

- In Windows, open the Control Panel, Administrative Tools, then Services (this may depend on your operating system).
- Locate the service called Serverengine_UUID , where UUID is a series of characters that depend on the machine where the software is installed.
- Right-click on the service and select *Properties*.
- In the *Connection* tab, define the account name and password that the service should use. This can be a local account on the computer or an account on a Windows Domain. The account must have administrative access on the machine. It should also correspond to the user account set up in *PReS Worfklow*.

The Importance of User Credentials on Installation and Running

OL Connect and required credentials depends heavily on the Connect component and respective tasks and what sort of user credentials are needed.

First of all, it is important to distinguish between installation and run-time

Installation

The Connect installer puts all required files, folders, registry entries and much more to their correct places and locations. As many of these locations are protected against malicious accesses, that very user under whose context the Connect installation is started and running, needs very extensive rights on the respective computer. This user must belong to the Local Administrators group on that machine. Here are some required capabilities, this user:

- Must be able to write into the "Programs" folder.
- Must be allowed to check for existing certificates and must also be allowed to install new ones into the global certificate store on that machine.
- Must be able to write into HKLM and any subtree of it in the registry.

- Must be able to INSTALL, START and RUN services and also to MODIFY service settings.
- Must be known in the network the machine belongs to and must also need to be able to use shared network resources like shared drives and/or printers etc.

This list may not be complete, but it gives the extent of the requirements. Generally, the local administrator of the machine will have all these credentials, but there may exist network restrictions and policies, which will block one or more of these capabilities. In such cases, the respective network administrator should provide a valid user account for the installation.

User Account

The user account shall be used to later RUN one of the Connect Server flavors (Server or Server Extension). This dedicated user account has to be entered on the respective installer dialog page and must be allowed to START, STOP and RUN services on this machine. This is different from the credentials of the installation user account, which additionally requires the right to INSTALL services. Please be aware of this fact!

Additionally, the Server user must be able to access any network resources that are required for OL Connect to function properly. This includes e.g. additional drives, printers, scanners, other computers and, where appropriate, internet resources, URLs, mail servers, FTP servers, database servers and everything else planned to be used for the intended operation of Connect. The Server user is the run-time user.

Connect Components

Usually, a standard end user will only be facing Connect Designer and maybe the License Activation Tool. Designer this does not require administrator rights. Either everything required to create documents or also to run some tasks will be already available (installed by the installer) or be accessible in a way, where no specific credentials are required. However some tasks like starting an email campaign will possibly require a respective account at a mail server. But this has generally nothing to do with the credentials of the Designer user.

Activation Tool

To run the Software Activation Tool, administrator rights are required because this tool needs to write the license file in one of the protected folders of Windows. The tool will however allow to restart it with respective credentials if required.

MySQL

MySQL database service is installed by the install user (thus again the requirement of installing, starting, running and modifying services). Once running it will just work.

Merge and Weaver Engines

These components do run under the Designer (if only Designer is installed) or the Server / Extension service(s) and inherit the rights of their parent application.

Server (Extension) Configuration Tool

This component needs to access the settings of the Server. As these are stored and read by the Server, it should be clear that the user used to run the Configuration tool should be the same as the Server Service user as explained above.

Server Settings

This chapter describes the different considerations that are important in regards to the use of PReS Connect Server.

- "Server Security Settings" below
- "Server Extension Settings" on the facing page

Server Security Settings

This dialog controls the security settings for external applications connecting to the PReS Connect Server, such as PReS Workflow or scripts communicating through the REST API.

> **Warning**
>
> It is **highly recommended** to keep security enabled and change the password on any server that accessible from the web. If these precautions are not taken, data saved in the server may be accessible from the outside!

- **Enable server security**: Enable to add authentication to the REST server. When disabled, a username and password is not required to make REST request, and tasks in PReS Workflow does not require them in the Proxy tab.

- **Administrator's username**: Enter the username for the server security. The default username is **ol-admin**.
- **Administrator's password**: Enter a password for the server security. The default password is **secret**.
- **Confirm password**: Re-enter the password for the server security.
- **Default session length (min)**: Enter a session time (in minutes) that the authentication stays valid for the requested process. This can reduce the number of requests to the server since an authentication request is not necessary during the session.

Server Extension Settings

This dialog controls the different settings for the PReS Connect Server Extension.

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

The following Preferences pages are available:

- "Cleanup Service preferences" on page 457
- "Server Extension Scheduling Preferences" on page 36 (these are different in the Server Extension preferences)
    - Merge Engine Scheduling
    - Weaver Engine Scheduling
- "Server Security Settings" on the previous page

Server Clustering

Server Clustering, available in PReS Connect, enhances the processing capabilities of PReS Connect Server by load-balancing jobs between the main Server module (master) and one or more Server Extension installations.

Setting up Server Clustering requires two or more installations of PReS Connect on separate machines. The Master server is setup by installing the PReS Connect Server module during the Installation Wizard, while the Slave Server is setup by installing the PReS Connect Server Extension module instead.

Quick Howto

1. Install the Master server (PReS Connect Server module), making sure to select the MySQL module.
2. Set the appropriate bindings in MySQL's my.ini file on the Master server.
3. Grant access to the MySQL root user for the appropriate IP range on the Master server.
4. Restart the MySQL Service on the Master server.
5. Install Slave servers (PReS Connect Server Extension module).
6. Install the license on the Master server (a Performance Pack license is required).
7. Set the preferences for the engines (see Scheduling Preferences) on both the Master and Slaves
8. Install the license on the Slave servers
9. Restart the Master server then, once restarted, restart the Slave servers.

What if MySQL is not on the Master server?

It is possible to setup clustering with a MySQL instance that is on a Slave server instead of on the master. In this case, the Slave server must be installed with the Server Extension and MySQL modules, the MySQL instance configured (steps 2-4 above) then the master and other slaves can be installed. The remainder of the instructions remain the same.

It is also possible to setup clustering with MySQL being installed completely separately from PReS Connect, such as using an existing MySQL instance. In this case, the instructions for the bind address must be followed, but the user does not have to be root. A user for MySQL must, however, be created and have full access (GRANT ALL PRIVILEGES) to a database called "objectiflune" that can be created before Connect is installed.

Binding and Root access on the Master server

- The MySQL server's binding must be set to accept connections from the slave servers. To do this, open C:\Program Files\Objectif Lune\OL Connect\MySQL Product\MySQL\my.ini in a text editor and change the line `bind-address= 127.0.0.1` to `bind-address=0.0.0.0`.
  - Once the changes have been made and saved you need to restart the MySQL services.
- Access must be granted to the `root` user on the IPs from which the Slave server will connect:
  - Open a Command Prompt in the following folder: C:\Program Files\Objectif Lune\OL Connect\MySQL Product\MySQL\bin (tip: navigate to the folder, SHIFT+Right-click and select "Open a command prompt here!).

- Type in the following command to connect to the database, where <password> is your MySQL password (by default it is **admin**):
  `mysql --user=root --password=<password> objectiflune`
- You should see the prompt become mysql>. Here, type the command to allow the "root" user to be accessed from a specific IP subnet range. For example, to accept communication on 192.168.*.*, use: `GRANT ALL PRIVILEGES ON objectiflune TO 'root'@'192.168.0.0/255.255.0.0' IDENTIFIED BY 'password'` , where `password` is the one provided during installation. (ref: http://dev.mysql.com/doc/mysql-security-excerpt/5.5/en/adding-users.html)

IP Subnets understanding is beyond the scope of this documentation. If you want to learn more, please see the Subnetwork article on Wikipedia.

Clustering Preferences and Setup

When server extensions are installed and connected to a Master, the following options and settings change in availability or behavior:

- In the Scheduling Preferences of the Slave, both "Maximum Records" are ignored. Scheduling is handled by the Master.
- The "Expected Remote Merge Engine" and "Expected Remote Weaver Engine" in Merge Engine Scheduling and Weaver Engine Scheduling respectively, on the Master, should each equal the *total number of engines in all the slaves combined*.
  - For example, the Expected Remote Merge Engine on the Master should equal the total of "Local Engines Launched" for each slave.
  - If the number of expected remote engines is *lower* than the actual number, performance will not be optimal.
  - If the number of expected remote engines is *higher* than the actual number, jobs may fail and not complete.
- Cleanup Service requires special configuration on Clustering setups:
  - Cleanup service should not run simultaneously on all machines (staggered cleanup). Doing so may cause jobs not to be processed since all servers are busy.
  - Only the machine where the MySQL Server product is installed should attempt to cleanup database items. Essentially server that do not have MySQL should only run Orphan File Cleanup.

Server Extension Scheduling Preferences

The Server Extension Scheduling Preferences define the connection settings for the server extension (slave) and the main Server module (master).

- **Location of the master server**: Enter the location and port of the main Server module in the *hostname:port* format. For example, *192.168.100.123:9340* or *connect-master:9340*.
- **Username**: Enter the username expected by the OL Connect Server.
- **Password**: Enter the password expected by the OL Connect Server for the above username.
- Note that **Maximum records in a small job** and **Minimum records in a large job**:are note used in Server Extensions. All server scheduling is handled by the Master.

DataMapper Module

The DataMapper Module is a tool that helps in the preparation of data for a unified data model. You can then use that Data Model in the Designer module.

The tool does this through the use of a data mapping workflow consisting of multiple steps which extract data from each Source Record of a Data Source and places this data in a Data Model. This Data Model contains all the necessary information for the creation of a template in the Designer module. (see Data Model for more information)

Basics

Here is the procedure for a Data Mapping configuration that includes the steps required for a data extraction from a data source:

1. **Create a new Data Mapping configuration**.
   First, you create a new Data Mapping configuration manually from a file or by automatically using a wizard. In this first step, you select a Data Source. See Creating a new Data Mapping configuration.
2. **Configure settings for the data source**.
   In this step, you configure how the Data Sample is read by the DataMapper so it can delimit each record in the file (using Delimiters). See Data Source.
3. **Configure the data extraction workflow**.
   Here you configure the workflow steps that will be required to extract the data from the Data source to the Data Model. This way, data will be converted and prepared to be used by the **Designer** module. To learn more, see Data Extraction.
4. **Save the Data Mapping configuration**.
   Now it's time to save a copy of a data mapping configuration under the name of your choice. To learn more, see Saving A Data Mapping configuration.

Features

The DataMapper Module is a tool that extract data and prepares it for use in the Designer module. The tool does this through the use of a **Data Mapping Workflow** consisting of multiple **Steps** which process and extract data from each Source Record of a Data Source and stores it in a Record Set.

Data Mapping Configuration
A Data Mapping Configuration file can refer to either the current active configuration in the DataMapper, or a file on disk containing the information necessary for data mapping. It contains the extraction workflow (steps), Delimiter and Boundary Settings, and any imported Data Samples.

Data Mapping Workflow
The data mapping workflow is a series of Steps inside of a data mapping configuration that extract the appropriate data from the Source Record and places it into the Record (see Data Model).

The data mapping workflow always starts with the **Preprocessor** step and ends with the **Postprocessor** step and can contain as many of the different steps available in the DataMapper.

Data Model
A Data Model is the format into which Records are saved. It is also a file format that can be exchanged between different parts of the PReS Connect solution.

Data Source (Settings)
A data source is simply the source of the data. It can be a file (CSV, PDF, TXT, XML) or a particular database. The data might be located on the same computer as the program, or on another computer somewhere on a network.

The first step to do before creating a Data Mapping configuration is to set up **Input Data** (setting the delimiters) and **Boundaries** (setting the trigger) in the **Settings** pane.

Data Mapping Configuration

A Data Mapping Configuration file can refer to either the current active configuration in the DataMapper, or a file on disk containing the information necessary for data mapping. It contains the extraction workflow (steps), Delimiter and Boundary Settings, and any imported Data Samples.

Creating A New Data Mapping Configuration

You can create a new Data Mapping configuration from a data file by using a wizard or manually.

From a File

You can create a new data mapping configuration manually under the **From a file** section. As opposed to using a wizard, you must configure all settings in the **Settings** pane of the DataMapper interface. Please refer to Configuring Settings For The Data Source for more information.

A CSV File

To create a Data Mapping from a CSV file, use the following steps:

From the Welcome screen

1. Open the PReS Connect **Welcome** page by clicking the icon at the top right or select the **Help** menu and then **Welcome**.
2. Click **Create a New Configuration**.
3. From the **From a file** section, select **CSV**.
4. Click the **Browse** button and open the CSV file you want to work with.
5. Click **Finish**.

From the File menu

1. Click the **File** menu and select **New**.
2. Click the **Data mapping Configuration** drop-down and select **Files** and then **CSV File**.
3. Click **Next**.
4. Click the **Browse** button and open the CSV file you want to work with.
5. Click **Finish**.

A MS-Access File

To create a Data Mapping from a Microsoft Access database file, use the following steps:

From the Welcome screen

1. Open the PReS Connect **Welcome** page by clicking the icon at the top right or select the **Help** menu and then **Welcome**.
2. Click **Create a New Configuration**.
3. From the **From a file** section, select **MS Access**.
4. Click the **Browse** button and open the database file you want to work with.
5. Enter the password if needed and click **Next**.

From the File menu

1. Click the **File** menu and select **New**.
2. Click the **Data mapping Configuration** drop-down and select **Files** and then **Microsoft Access**.
3. Click **Next**.
4. Click the **Browse** button and open the database file you want to work with.
5. Enter the password if needed and click **Next**.

A PDF File

To create a PDF/VT file data mapping configuration, use the following steps:

From the Welcome screen

1. Open the PReS Connect **Welcome** page by clicking the icon at the top right or select the **Help** menu and then **Welcome**.
2. Click **Create a New Configuration**.
3. From the **From a file** section, select PDF/VT.
4. Click the **Browse** button and open the PDF/VT file you want to work with.
5. Click **Finish** to close the dialog and open the actual Data Mapping configuration).

From the File menu

1. Click the **File** menu and select **New**.
2. Click the **Data mapping Configuration** drop-down and select **Files** and then **PDF/PS/PCL/AFP File**.
3. Click **Next**.
4. Click the **Browse** button and open the PDF/VT file you want to work with.
5. Click **Finish** to close the dialog and open the actual Data Mapping configuration).

> **Note**
>
> PCL and PostScript (PS) files are automatically converted to PDF files before showing that PDF in the Data Viewer. This happens once when opening the file, but in automation happens for every file. Depending on the processing power available, this may influence the processing speed.

A Text File

To create a TXT file data mapping configuration, use the following steps:

From the Welcome screen

1. Open the PReS Connect **Welcome** page by clicking the icon at the top right or select the **Help** menu and then **Welcome**.
2. Click **Create a New Configuration**.
3. From the **From a file** section, select Text.
4. Click the **Browse** button and open the text file you want to work with.
5. Click **Finish** to close the dialog and open the actual Data Mapping configuration).

From the File menu

1. Click the **File** menu and select **New**
2. Click the **Data mapping Configuration** drop-down and select **Files** and then **Text File**.
3. Click **Next**.
4. Click the **Browse** button and open the text file you want to work with.
5. Click **Finish** to close the dialog and open the actual Data Mapping configuration).

A XML File

To create a XML File data mapping configuration, use the following steps:

From the Welcome screen

1. Open the PReS Connect **Welcome** page by clicking the icon at the top right or select the **Help** menu and then **Welcome**.
2. Click **Create a New Configuration**.
3. From the **From a file** section, select XML.
4. Click the **Browse** button and open the XML file you want to work with.
5. Click **Finish** to close the dialog and open the Data Mapping configuration.

From the File menu

1. Click the **File** menu and select **New**.
2. Click the **Data mapping Configuration** drop-down and select **Files** and then **XML File**.
3. Click **Next**.
4. Click the **Browse** button and open the XML file you want to work with.
5. Click **Finish** to close the dialog and open the Data Mapping configuration.

Using a Wizard

The Data Mapping Wizard simplifies extracting records. When using a Data Mapping Wizard, all the fields are automatically extracted inside the record. The Data Mapping Wizard also preconfigure the settings and boundaries of the data source using common values.

For A CSV File

To create a Data Mapping from a CSV file using the wizard, use the following steps:

From the Welcome screen

1. Open the PReS Connect **Welcome** page by clicking the icon at the top right or select the **Help** menu and then **Welcome**.
2. Click **Create a New Configuration**.
3. From the **Using a wizard** section, select **CSV**.
4. Click the **Browse** button and open the CSV file you want to work with.
   - Take a look at the **Preview** box content to ensure that the file is the right one and the encoding correctly reads the data.
   - Click **Next**.
5. From the **Select a CSV Configuration** dialog, choose the proper settings:

   Note

   These settings are generally detected automatically.

   - **Encoding**: Choose the correct encoding to read the file.
   - **Separator**: Defines what character separates each fields in the file.
   - **Comment Delimiter**: Defines what character starts a comment line.
   - **Text Delimiter**: Defines what character surrounds text fields in the file, preventing the Field Delimiter from being interpreted within those text delimiters.
   - **Ignore unparseable lines**: Ignores any line that does not correspond to the settings above.
   - **First row contains field names**: Uses the first line of the CSV as headers, which automatically names all extracted fields.
6. Verify that the data is read properly and click **Finish**.

From the File menu

1. Click the **File** menu and select **New**.Click the **Data mapping Data mapping Wizards** drop-down and select **From CSV File**.
2. Click **Next**.
3. Click the **Browse** button and open the CSV file you want to work with.
   - Take a look at the **Preview** box content to ensure that the file is the right one and the encoding correctly reads the data.
   - Click **Next**.
4. From the **Select a CSV Configuration** dialog, choose the proper settings:

   > **Note**
   >
   > These settings are generally detected automatically.

   - **Encoding**: Choose the correct encoding to read the file.
   - **Separator**: Defines what character separates each fields in the file.
   - **Comment Delimiter**: Defines what character starts a comment line.
   - **Text Delimiter**: Defines what character surrounds text fields in the file, preventing the Field Delimiter from being interpreted within those text delimiters.
   - **Ignore unparseable lines**: Ignores any line that does not correspond to the settings above.
   - **First row contains field names**: Uses the first line of the CSV as headers, which automatically names all extracted fields.
5. Verify that the data is read properly and click **Finish**.

For a Database File

To create a Data Mapping from a Database file using the wizard, use the following steps:

From the Welcome screen

1. Open the PReS Connect **Welcome** page by clicking the icon at the top right or select the **Help** menu and then **Welcome**.
2. Click **Create a New Configuration**.
3. From the **Using a wizard** section, select Database.

4. Use the drop-down to select the Database type.
5. Click **Next**.

From the File menu

1. Click the **File** menu and select **New**.Click the **Data mapping Data mapping Wizards** drop-down and select **From databases**.
2. Click **Next**.
3. Use the drop-down to select the Database type.
4. Click **Next**.

Now set the following properties according to the selected database:

MySQL

- **Server**: Enter the server address for the MySQL database.
- **Port**: Enter the port to communicate with the 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 MySQL server and specified database. The user only requires *Read*access to the database.
- **Password**: Enter the password that matches the user name above.
- **Table name**: The selected database is a set of related tables composed of rows and columns corresponding respectively to source records and fields. Select a table from which you want to extract data.
- **Encoding**: Choose the correct encoding to read the file.
- Click **Finish** to close the dialog and open the actual Data Mapping configuration.

Microsoft Access

- Click the **Browse** button and open the database file you want to work with.
- **Password**: Enter a password if one is required.
- Click **Next**.
- **Table name**: The selected database is a set of related tables composed of rows and columns corresponding respectively to source records and fields. Select a table from which you want to extract data.
- **Encoding**: Choose the correct encoding to read the file.
- Click **Finish** to close the dialog and open the actual Data Mapping configuration.

SQL Server

- **Server**: Enter the server address for the SQL Server database.
- **Port**: Enter the port to communicate with the SQL Server 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 SQL Server server and specified database. The user only requires *Read*access to the database.
- **Password**: Enter the password that matches the user name above.
- **Table name**: The selected database is a set of related tables composed of rows and columns corresponding respectively to source records and fields. Select a table from which you want to extract data.
- **Encoding**: Choose the correct encoding to read the file.
- Click **Finish** to close the dialog and open the actual Data Mapping configuration.

ODBC DataSource

- **ODBC Source**: Use the drop-down to select an ODBC System Data Source. This must be a data source that has been configured in the 64-bit ODBC Data Source Administrator, as PReS Connect is a 64-bit application and thus cannot access 32-bit data sources.
- **This ODBC source is MSSQL**: Check this option if the ODBC source is MSSQL (SQL Server). The options below appear under **MSSQL-ODBC advanced configuration**:
    - **Windows authentication**: Select to use the Windows User name and Password that are used by the Connect Service.
    - **SQL Server authentication**: Select to use the User name and Password set below to connect to the SQL Server:
        - **User name**: Enter the SQL Server user name.
        - **Password**: Enter the password for the above user name.
    - Click **Next**.
    - Click **Finish** to close the dialog and open the actual Data Mapping configuration.

JDBC

> **Note**
>
> Since JDBC can connect to multiple types of databases, a specific database driver and path to

> this driver's JAR file must be specified.

- **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.
- **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 server and specified database. The user only requires **Read**access to the database.
- **Password**: Enter the password that matches the user name above.
- **Advanced mode**: Check to enable the **Connection String** field to manually enter the database connection string.
- **Connection string**: Type or copy in your connection string.
- Click **Next**
- Click **Finish** to close the dialog and open the actual Data Mapping configuration.

Oracle

- **Server**: Enter the server address for the Oracle database.
- **Port**: Enter the port to communicate with the Oracle 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 Oracle server and specified database. The user only requires *Read*access to the database.
- **Password**: Enter the password that matches the user name above.
- **Table name**: The selected database is a set of related tables composed of rows and columns corresponding respectively to source records and fields. Select a table from which you want to extract data.
- **Encoding**: Choose the correct encoding to read the file.
- Click **Finish** to close the dialog and open the actual Data Mapping configuration.

For a PDF File

To create a PDF/VT file data mapping configuration using the wizard, use the following steps:

From the Welcome screen

1. Open the PReS Connect **Welcome** page by clicking the icon at the top right or select the **Help** menu and then **Welcome**.
2. Click **Create a New Configuration**.
3. From the **Using a wizard** section, select **PDF/VT**.
4. Click the **Browse** button and open the PDF/VT file you want to work with. Click **Next**.
5. In the **Metadata** page, select the following options:
    - **Metadata record levels**: Use the drop-down to select what level in the metadata defines a Source Record.
    - **Field List**: This list displays all fields in all levels of the PDF/VT metadata.
        - **Checkmark**: Check any field to add it to the extraction.
        - **Record Level**: Displays the level on which the field is located.
        - **Property name**: Displays the field names to extract.
6. Click **Finish** to close the dialog and open the actual Data Mapping configuration).

From the File menu

1. Click the **File** menu and select **New**.
2. Click the **Data mapping Data mapping Wizards** drop-down and select **From PDF/VT or AFP**.
3. Click **Next**.
4. Click the **Browse** button and open the PDF/VT file you want to work with. Click **Next**.
5. In the **Metadata** page, select the following options:
    - **Metadata record levels**: Use the drop-down to select what level in the metadata defines a Source Record.
    - **Field List**: This list displays all fields in all levels of the PDF/VT metadata.
        - **Checkmark**: Check any field to add it to the extraction.
        - **Record Level**: Displays the level on which the field is located.
        - **Property name**: Displays the field names to extract.
6. Click **Finish** to close the dialog and open the actual Data Mapping configuration).

From a XML File

To create a XML file data mapping configuration, use the following steps:

From the Welcome screen

1. Open the PReS Connect **Welcome** page by clicking the icon at the top right or select the **Help** menu and then **Welcome**.
2. Click **Create a New Configuration**.
3. From the **Using a wizard** section, select XML.
4. Click the **Browse** button and open the XML file you want to work with. Click **Next**.
5. In the **Select split level and trigger type** page, select the following options:
    - **XML Elements**: A list of node elements that have children nodes. Select the level in the data that will define the Source Record (for example Invoice, Customer ID, Item...etc as opposed to Last name, Due date...etc).
    - **Trigger**: According to the node element path selected in the **XML Elements** field, select **On element** to create a record in the Data mapping for each occurrence of the element. Or you can select **On change** to create a record each time the element is different.
6. Click **Finish** to close the dialog and open the Data Mapping configuration.

From the File menu

1. Click the **File** menu and select **New**.Click the **Data mapping Data mapping Wizards** drop-down and select **From XML File**.
2. Click **Next**.
3. Click the **Browse** button and open the XML file you want to work with. Click **Next**.
4. In the **Select split level and trigger type** page, select the following options:
    - **XML Elements**: A list of node elements that have children nodes. Select the level in the data that will define the Source Record (for example Invoice, Customer ID, Item...etc as opposed to Last name, Due date...etc).
    - **Trigger**: According to the node element path selected in the **XML Elements** field, select **On element** to create a record in the Data mapping for each occurrence of the element. Or you can select **On change** to create a record each time the element is different.
5. Click **Finish** to close the dialog and open the Data Mapping configuration.

To Generate Counter

The Generate Counter wizard creates a record set where each record contains a "counter" field contains the current counter value for each record, starting and stopping at set values and incremented by a set value also. It is useful for numbered tickets or any other template requiring sequential numbers.

- **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. 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.

Opening a Data Mapping Configuration

There are two ways you can open a data mapping configuration:

- In the Menus, click on **File**, **Open**, ensure that the file type is either DataMapper files or Connect files. Browse to the configuration file to open, select it and click **Open**.
- In the Menus, click on**File**, **Open Recent**, and select one of the recently opened configuration files.

Saving a Data Mapping Configuration

To save a data mapping configuration:

1. In the Menus, click on **File**, then **Save**, *or* click the **Save** button in theToolbars.
2. If the data mapping configuration has never been saved, browse to the location where the data mapping configuration should be saved and type a name, then click **Save**.

To save a **copy** of a data mapping configuration under a different name:

1. In the Menus, click on **File**, then **Save As**.
2. Browse to the location where the data mapping configuration should be saved and type a name, then click **Save**.

Data Mapping Workflow

A data mapping workflow is a series of steps used to process and extract the appropriate data from the Source Record and store it into the Record (see Data Model). With the data model, it is what makes a data mapping configuration (See Data Mapping Configuration).

A data mapping workflow always starts with the **Preprocessor** step and ends with the **Postprocessor** step. It can contain as many steps as needed for extracting the required data.

When working with a data mapping workflow, you need to consider the following:

- **Promotional versus transactional data**: Depending on the type of data, whether it is promotional or transactional, you will use a different extraction workflow. See About Promotional and Transactional Data for more information.
- **Steps**: To extract data from the data sample, you use different steps that make up the extraction process. See Steps for more information.

About Promotional and Transactional Data

Promotional Data is traditionally defined as the unique fields found in each record. For instance, a postal address, a date, or customer number could all be elements of promotional data found in each record.

Transactional data is usually defined as promotional data plus a varying number of fields that represent the elements of a transaction. While the structure is identical for each record, the number of repeated elements varies from one record to the next. For instance, lines of items on an invoice or monthly usage of electricity would be part of the transactional data.

Data Extraction

The following example explains in details how to perform a data extraction for each different data source file types and the **Steps** that you can use to reach that goal. Delimiters and Boundaries must be properly configured beforehand (see Configuring Settings for more information).

The Preprocessor and the Postprocessor

Data processors allow the application to perform actions on the data file itself before it is handed over to the Data Mapping workflow (preprocessors) and after the Data Mapping workflow has completed (postprocessors).

To configure settings for the **Preprocessor** step, see Preprocessor Step Properties.

To configure settings for the **Postprocessor** step, see Postprocessor Step Properties.

The Extraction

The **Extract** step is the heart of the DataMapper software. It takes information from the Data Sample and places it in the Extracted Record within the Record Set.

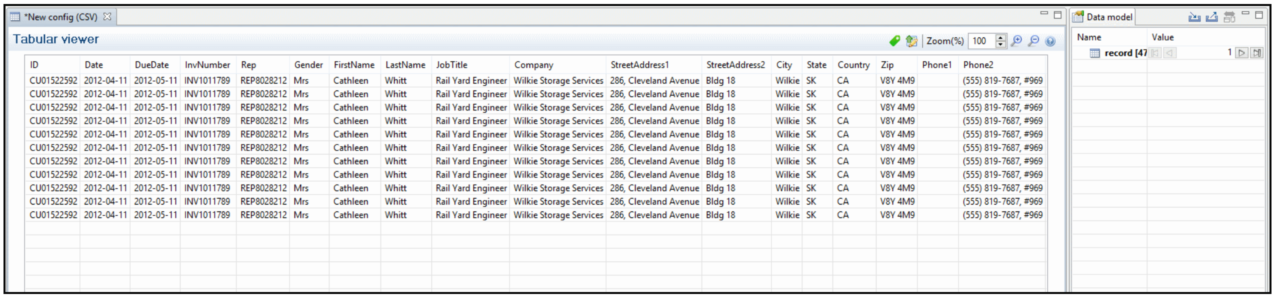
To configure settings for the **Extract** step, see Extract Step Properties.

For more information about Promotional and Transactional data, please refer to About Promotional and Transactional Data.

From a CSV file or a Database

Extracting Promotional Data

1. From the Viewer pane, select the fields that contain the customer and invoice information. For more information about how to select data in a CSV file, please refer to Data Selection/From a CSV File.
2. Drag & drop the selected fields into the Data Model pane.



ID	Date	DueDate	InvNumber	Rep	Gender	FirstName	LastName	JobTitle	Company	StreetAddress1	StreetAddress2	City	State	Country	Zip	Phone1	Phone2
CU01522592	2012-04-11	2012-05-11	INV1011789	REP8028212	Mrs	Cathleen	Whitt	Rail Yard Engineer	Wilkie Storage Services	286, Cleveland Avenue	Bldg 18	Wilkie	SK	CA	V8Y 4M9		(555) 819-7687, #969
CU01522592	2012-04-11	2012-05-11	INV1011789	REP8028212	Mrs	Cathleen	Whitt	Rail Yard Engineer	Wilkie Storage Services	286, Cleveland Avenue	Bldg 18	Wilkie	SK	CA	V8Y 4M9		(555) 819-7687, #969
CU01522592	2012-04-11	2012-05-11	INV1011789	REP8028212	Mrs	Cathleen	Whitt	Rail Yard Engineer	Wilkie Storage Services	286, Cleveland Avenue	Bldg 18	Wilkie	SK	CA	V8Y 4M9		(555) 819-7687, #969
CU01522592	2012-04-11	2012-05-11	INV1011789	REP8028212	Mrs	Cathleen	Whitt	Rail Yard Engineer	Wilkie Storage Services	286, Cleveland Avenue	Bldg 18	Wilkie	SK	CA	V8Y 4M9		(555) 819-7687, #969
CU01522592	2012-04-11	2012-05-11	INV1011789	REP8028212	Mrs	Cathleen	Whitt	Rail Yard Engineer	Wilkie Storage Services	286, Cleveland Avenue	Bldg 18	Wilkie	SK	CA	V8Y 4M9		(555) 819-7687, #969
CU01522592	2012-04-11	2012-05-11	INV1011789	REP8028212	Mrs	Cathleen	Whitt	Rail Yard Engineer	Wilkie Storage Services	286, Cleveland Avenue	Bldg 18	Wilkie	SK	CA	V8Y 4M9		(555) 819-7687, #969
CU01522592	2012-04-11	2012-05-11	INV1011789	REP8028212	Mrs	Cathleen	Whitt	Rail Yard Engineer	Wilkie Storage Services	286, Cleveland Avenue	Bldg 18	Wilkie	SK	CA	V8Y 4M9		(555) 819-7687, #969
CU01522592	2012-04-11	2012-05-11	INV1011789	REP8028212	Mrs	Cathleen	Whitt	Rail Yard Engineer	Wilkie Storage Services	286, Cleveland Avenue	Bldg 18	Wilkie	SK	CA	V8Y 4M9		(555) 819-7687, #969
CU01522592	2012-04-11	2012-05-11	INV1011789	REP8028212	Mrs	Cathleen	Whitt	Rail Yard Engineer	Wilkie Storage Services	286, Cleveland Avenue	Bldg 18	Wilkie	SK	CA	V8Y 4M9		(555) 819-7687, #969
CU01522592	2012-04-11	2012-05-11	INV1011789	REP8028212	Mrs	Cathleen	Whitt	Rail Yard Engineer	Wilkie Storage Services	286, Cleveland Avenue	Bldg 18	Wilkie	SK	CA	V8Y 4M9		(555) 819-7687, #969
CU01522592	2012-04-11	2012-05-11	INV1011789	REP8028212	Mrs	Cathleen	Whitt	Rail Yard Engineer	Wilkie Storage Services	286, Cleveland Avenue	Bldg 18	Wilkie	SK	CA	V8Y 4M9		(555) 819-7687, #969

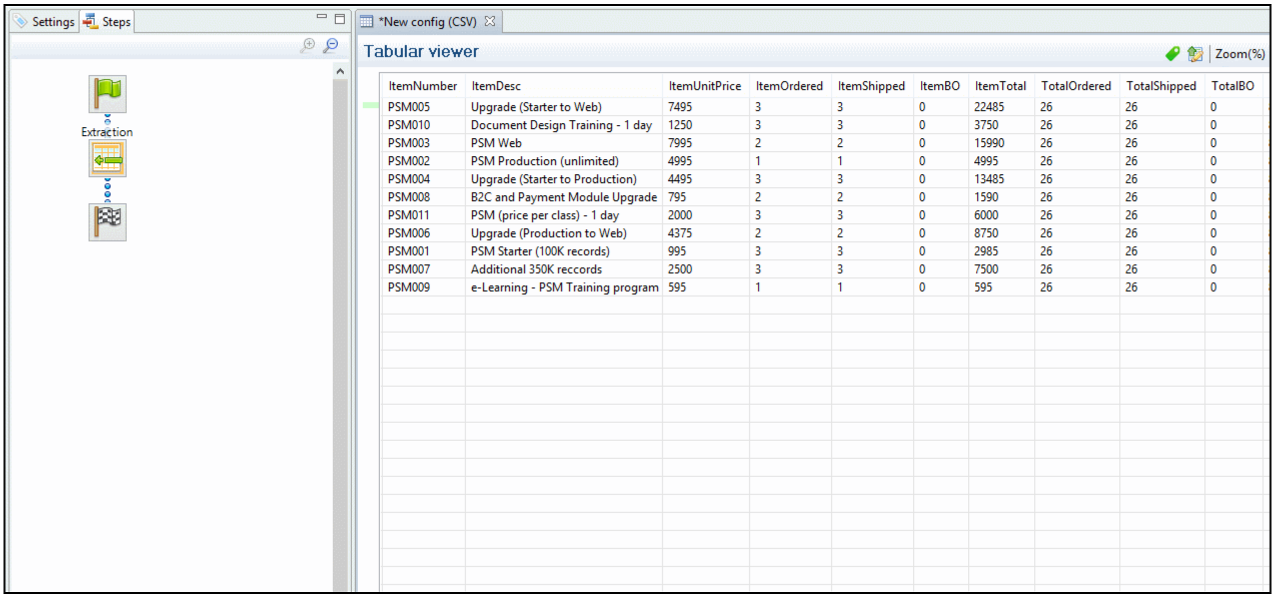
Note

Alternatively, you can simply right click on the selected fields and select **Add Extraction**.

Extracting Transactional Data

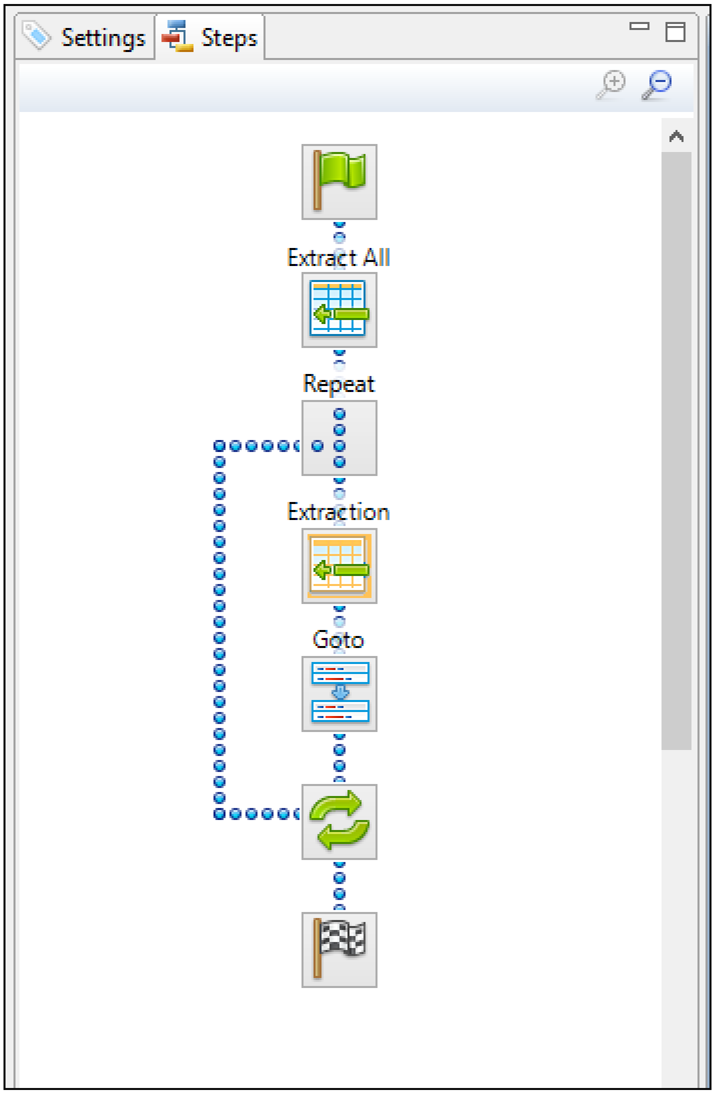
The Transactional Data (line items information) appears on multiple lines. You must create a loop on these lines to extract the items information. The line items are extracted in a detail table.

1. Select the fields that contain the first line item information.
2. Right click on this data selection and select **Add Repeat** .



ItemNumber	ItemDesc	ItemUnitPrice	ItemOrdered	ItemShipped	ItemBO	ItemTotal	TotalOrdered	TotalShipped	TotalBO
PSM005	Upgrade (Starter to Web)	7495	3	3	0	22485	26	26	0
PSM010	Document Design Training - 1 day	1250	3	3	0	3750	26	26	0
PSM003	PSM Web	7995	2	2	0	15990	26	26	0
PSM002	PSM Production (unlimited)	4995	1	1	0	4995	26	26	0
PSM004	Upgrade (Starter to Production)	4495	3	3	0	13485	26	26	0
PSM008	B2C and Payment Module Upgrade	795	2	2	0	1590	26	26	0
PSM011	PSM (price per class) - 1 day	2000	3	3	0	6000	26	26	0
PSM006	Upgrade (Production to Web)	4375	2	2	0	8750	26	26	0
PSM001	PSM Starter (100K records)	995	3	3	0	2985	26	26	0
PSM007	Additional 350K reccords	2500	3	3	0	7500	26	26	0
PSM009	e-Learning - PSM Training program	595	1	1	0	595	26	26	0

3. Right click again on this data selection and select **Add Extraction**. A new extraction step will be placed between the **Repeat** and the**Goto** steps.



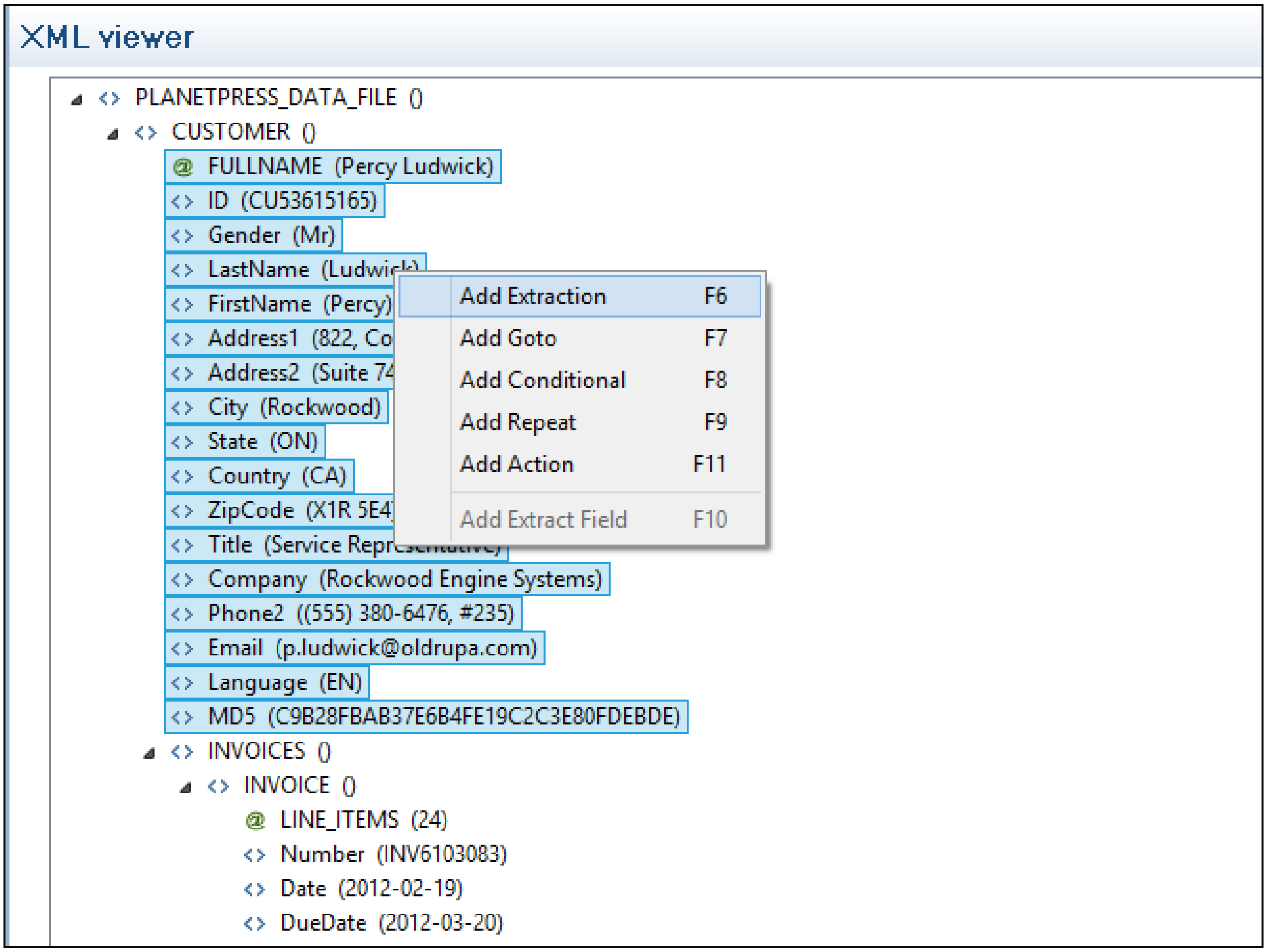
> **Note**
>
> You can also use the toolbar buttons instead of contextual menus, See Toolbars for more details.

From a XML file

Extracting Promotional Data

The Promotional Data (customer and the invoice information) is repeated for each item normally located at the top of the Source Record, before the ITEM information. It generally includes the name, the address, reference numbers, invoice information...etc.

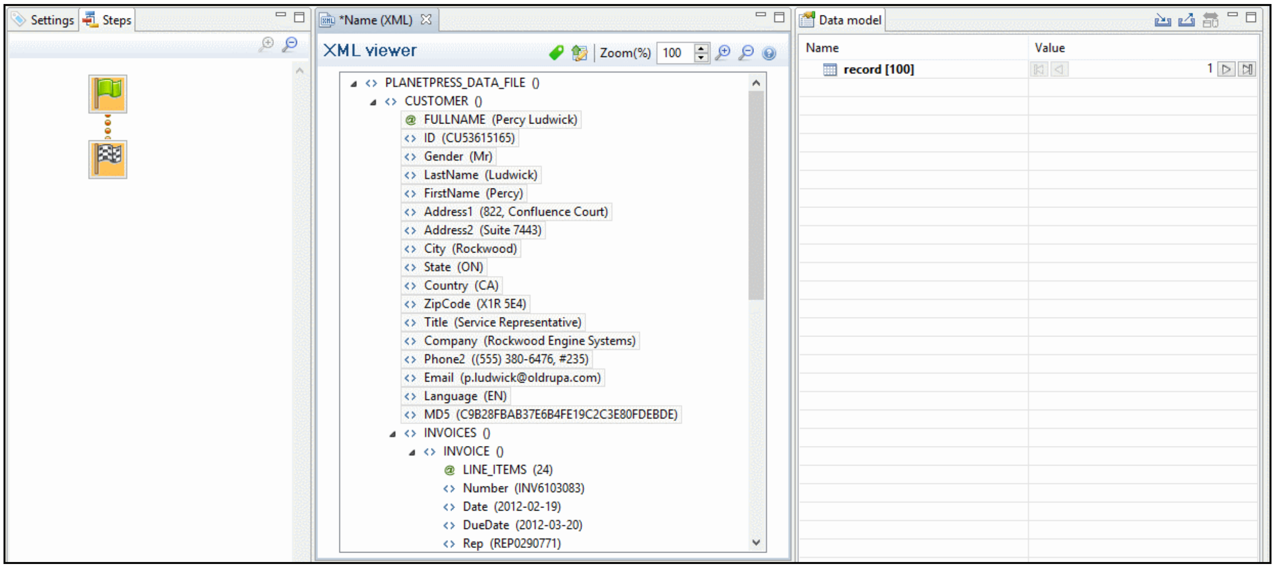
1. Select the children leaf nodes of the CUSTOMER() node.



2. Right click on the selected nodes and select **Add Extraction**.

Note

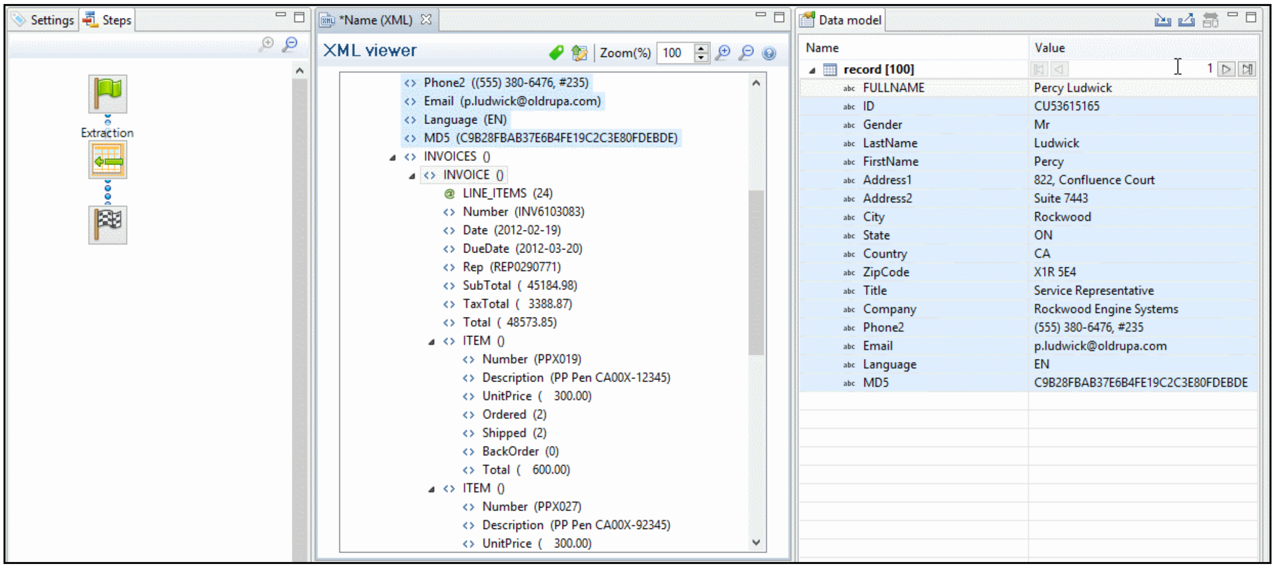
Alternatively, you can simply drag & drop the selected fields into the Data Model pane.



```
PLANETPRESS_DATA_FILE ()
  CUSTOMER ()
    FULLNAME (Percy Ludwick)
    ID (CU53615165)
    Gender (Mr)
    LastName (Ludwick)
    FirstName (Percy)
    Address1 (822, Confluence Court)
    Address2 (Suite 7443)
    City (Rockwood)
    State (ON)
    Country (CA)
    ZipCode (X1R 5E4)
    Title (Service Representative)
    Company (Rockwood Engine Systems)
    Phone2 ((555) 380-6476, #235)
    Email (p.ludwick@oldrupa.com)
    Language (EN)
    MD5 (C9B28FBAB37E6B4FE19C2C3E80FDEBDE)
    INVOICES ()
      INVOICE ()
        LINE_ITEMS (24)
        Number (INV6103083)
        Date (2012-02-19)
        DueDate (2012-03-20)
        Rep (REP0290771)
```

Name	Value
record [100]	1

3. Select the children leaf nodes of the INVOICE() node.
4. Right click on the selected nodes and select **Add Extraction**.



```
    Phone2 ((555) 380-6476, #235)
    Email (p.ludwick@oldrupa.com)
    Language (EN)
    MD5 (C9B28FBAB37E6B4FE19C2C3E80FDEBDE)
    INVOICES ()
      INVOICE ()
        LINE_ITEMS (24)
        Number (INV6103083)
        Date (2012-02-19)
        DueDate (2012-03-20)
        Rep (REP0290771)
        SubTotal ( 45184.98)
        TaxTotal ( 3388.87)
        Total ( 48573.85)
        ITEM ()
          Number (PPX019)
          Description (PP Pen CA00X-12345)
          UnitPrice (   300.00)
          Ordered (2)
          Shipped (2)
          BackOrder (0)
          Total (   600.00)
        ITEM ()
          Number (PPX027)
          Description (PP Pen CA00X-92345)
          UnitPrice (   300.00)
```

Name	Value
record [100]	1
FULLNAME	Percy Ludwick
ID	CU53615165
Gender	Mr
LastName	Ludwick
FirstName	Percy
Address1	822, Confluence Court
Address2	Suite 7443
City	Rockwood
State	ON
Country	CA
ZipCode	X1R 5E4
Title	Service Representative
Company	Rockwood Engine Systems
Phone2	(555) 380-6476, #235
Email	p.ludwick@oldrupa.com
Language	EN
MD5	C9B28FBAB37E6B4FE19C2C3E80FDEBDE

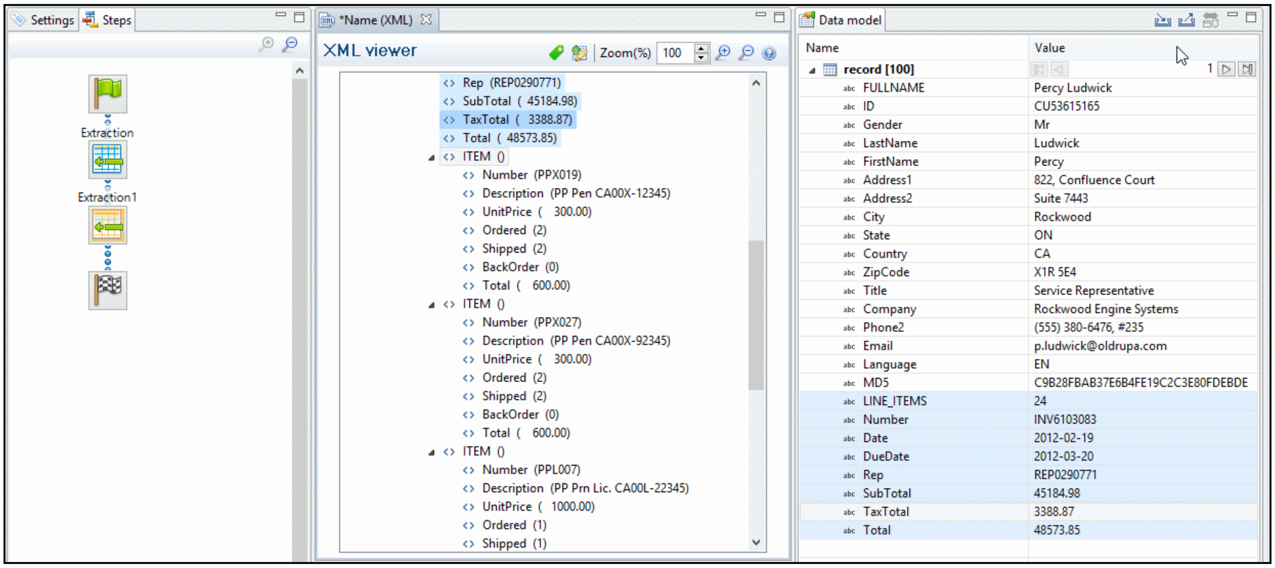
Extracting Transactional Data

The Transactional Data (line items information) appears on multiple lines. In the example below, this information appears under the parent node ITEM(). Each ITEM() node give information about one item. Create a loop on the ITEM() nodes to extract the items information.

1. Select the ITEM() node.
2. Right click on the selected node and select **Add Repeat**.

Note

By default, the **For Each** option is selected in the **Repeat type** option as shown in the**Step Properties** pane. It means that the loop will include each ITEM() node. In the **Collection** field, you will find the corresponding ITEM() node path.

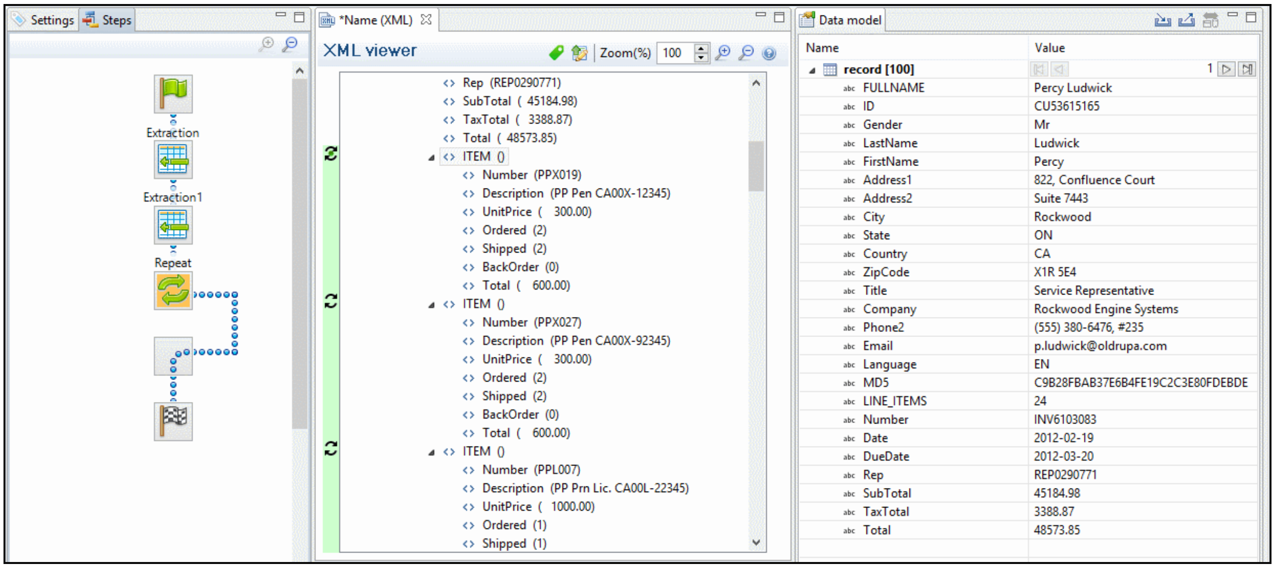


```
<> Rep (REP0290771)
<> SubTotal ( 45184.98)
<> TaxTotal (  3388.87)
<> Total ( 48573.85)
<> ITEM ()
    <> Number (PPX019)
    <> Description (PP Pen CA00X-12345)
    <> UnitPrice (   300.00)
    <> Ordered (2)
    <> Shipped (2)
    <> BackOrder (0)
    <> Total (   600.00)
<> ITEM ()
    <> Number (PPX027)
    <> Description (PP Pen CA00X-92345)
    <> UnitPrice (   300.00)
    <> Ordered (2)
    <> Shipped (2)
    <> BackOrder (0)
    <> Total (   600.00)
<> ITEM ()
    <> Number (PPL007)
    <> Description (PP Prn Lic. CA00L-22345)
    <> UnitPrice (  1000.00)
    <> Ordered (1)
    <> Shipped (1)
```

Name	Value
record [100]	1
FULLNAME	Percy Ludwick
ID	CU53615165
Gender	Mr
LastName	Ludwick
FirstName	Percy
Address1	822, Confluence Court
Address2	Suite 7443
City	Rockwood
State	ON
Country	CA
ZipCode	X1R 5E4
Title	Service Representative
Company	Rockwood Engine Systems
Phone2	(555) 380-6476, #235
Email	p.ludwick@oldrupa.com
Language	EN
MD5	C9B28FBAB37E6B4FE19C2C3E80FDEBDE
LINE_ITEMS	24
Number	INV6103083
Date	2012-02-19
DueDate	2012-03-20
Rep	REP0290771
SubTotal	45184.98
TaxTotal	3388.87
Total	48573.85

3. Select the children leaf nodes of the ITEM() node.

4. Right click on the selected nodes and select **Add Extraction**.



```
<> Rep  (REP0290771)
<> SubTotal  (  45184.98)
<> TaxTotal  (   3388.87)
<> Total  (  48573.85)
<> ITEM  ()
    <> Number  (PPX019)
    <> Description  (PP Pen CA00X-12345)
    <> UnitPrice  (    300.00)
    <> Ordered  (2)
    <> Shipped  (2)
    <> BackOrder  (0)
    <> Total  (    600.00)
<> ITEM  ()
    <> Number  (PPX027)
    <> Description  (PP Pen CA00X-92345)
    <> UnitPrice  (    300.00)
    <> Ordered  (2)
    <> Shipped  (2)
    <> BackOrder  (0)
    <> Total  (    600.00)
<> ITEM  ()
    <> Number  (PPL007)
    <> Description  (PP Prn Lic. CA00L-22345)
    <> UnitPrice  (   1000.00)
    <> Ordered  (1)
    <> Shipped  (1)
```

Name	Value
record [100]	1
FULLNAME	Percy Ludwick
ID	CU53615165
Gender	Mr
LastName	Ludwick
FirstName	Percy
Address1	822, Confluence Court
Address2	Suite 7443
City	Rockwood
State	ON
Country	CA
ZipCode	X1R 5E4
Title	Service Representative
Company	Rockwood Engine Systems
Phone2	(555) 380-6476, #235
Email	p.ludwick@oldrupa.com
Language	EN
MD5	C9B28FBAB37E6B4FE19C2C3E80FDEBDE
LINE_ITEMS	24
Number	INV6103083
Date	2012-02-19
DueDate	2012-03-20
Rep	REP0290771
SubTotal	45184.98
TaxTotal	3388.87
Total	48573.85

> **Note**
>
> You can also use the toolbar buttons instead of contextual menus, See Toolbars for more details.

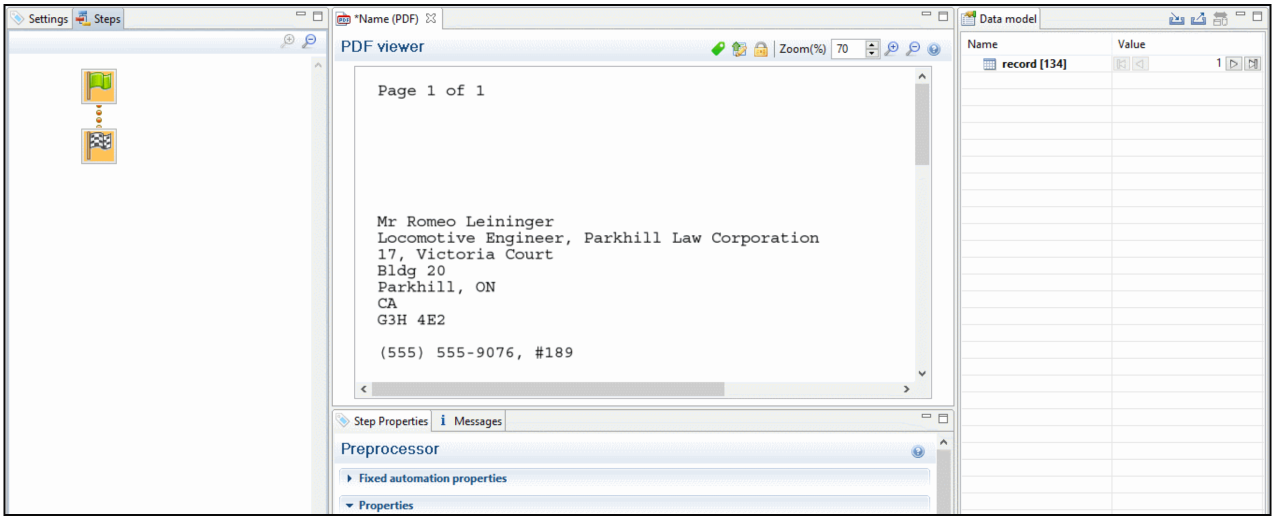
From a Text or a PDF file

Extracting Promotional Data

The Promotional Data (customer and the invoice information) is repeated for each item and normally located at the top of the Source Record. It generally includes the name, the address, reference numbers, invoice information...etc.

1. From the Viewer pane, select the customer and invoice information. Note that you can select multiple lines.
2. Right click on the selection data and select **Add Extraction** .

> **Note**
>
> Alternatively, you can simply drag & drop the selected fields into the Data Model pane.

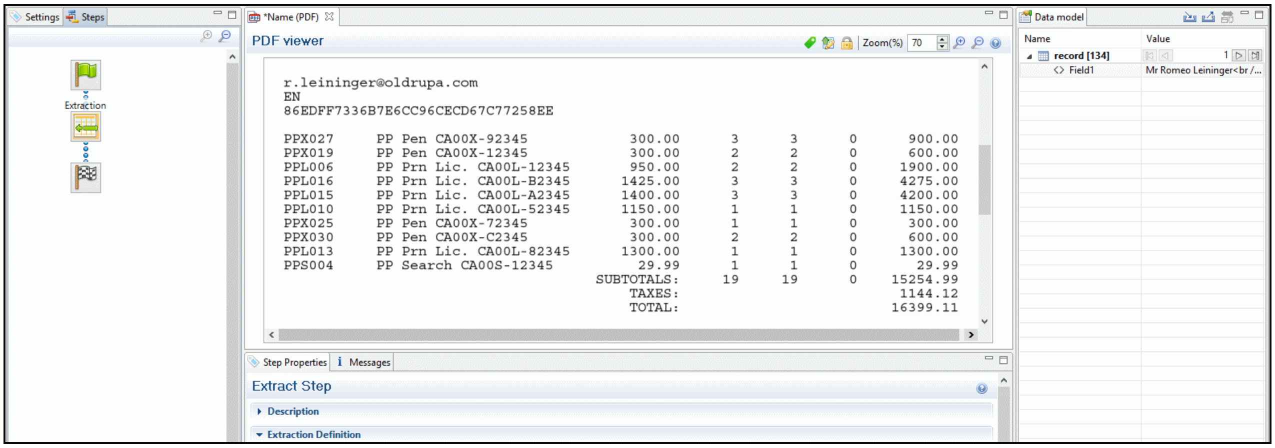


```
Page 1 of 1



Mr Romeo Leininger
Locomotive Engineer, Parkhill Law Corporation
17, Victoria Court
Bldg 20
Parkhill, ON
CA
G3H 4E2

(555) 555-9076, #189
```

Extracting Transactional Data

The Transactional Data (line items information) appears on multiple lines and pages. A loop has to be created on these lines to extract the items information. The line items are extracted in a detail table as below:

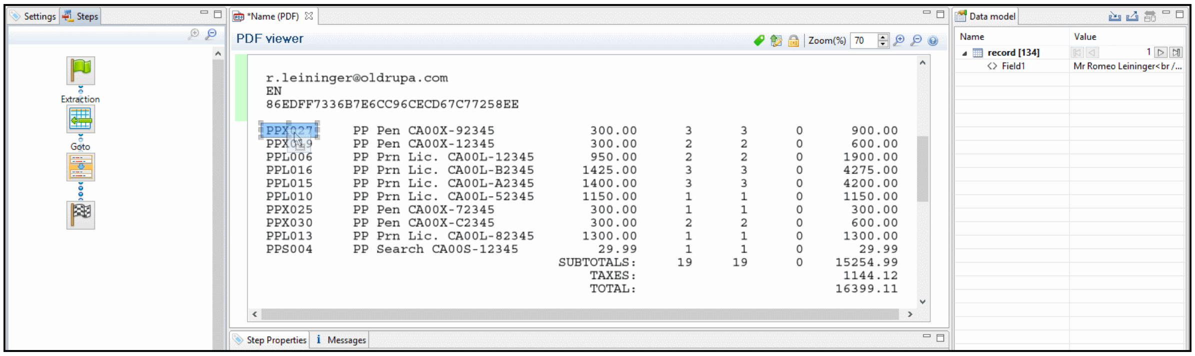
1. Select a simple data in the first line item. For example the product number.
2. Right click on the selection and select **Add Goto**. That moves the cursor at the beginning of the first line item.



```
r.leininger@oldrupa.com
EN
86EDFF7336B7E6CC96CECD67C77258EE

PPX027     PP Pen CA00X-92345          300.00     3      3      0      900.00
PPX019     PP Pen CA00X-12345          300.00     2      2      0      600.00
PPL006     PP Prn Lic. CA00L-12345     950.00     2      2      0     1900.00
PPL016     PP Prn Lic. CA00L-B2345    1425.00     3      3      0     4275.00
PPL015     PP Prn Lic. CA00L-A2345    1400.00     3      3      0     4200.00
PPL010     PP Prn Lic. CA00L-52345    1150.00     1      1      0     1150.00
PPX025     PP Pen CA00X-72345          300.00     1      1      0      300.00
PPX030     PP Pen CA00X-C2345          300.00     2      2      0      600.00
PPL013     PP Prn Lic. CA00L-82345    1300.00     1      1      0     1300.00
PPS004     PP Search CA00S-12345        29.99     1      1      0       29.99
                                  SUBTOTALS:     19     19      0    15254.99
                                      TAXES:                          1144.12
                                      TOTAL:                         16399.11
```

3. Add a loop to extract each item until the end of the line items. To stop the loop at the right place, you can use a text located at the end each record. For example, it can be a text string like SUBTOTALS, TOTAL, AMOUNT, ...etc. Now you can use that text as a condition to stop the loop at the end of the line items list. In that case:

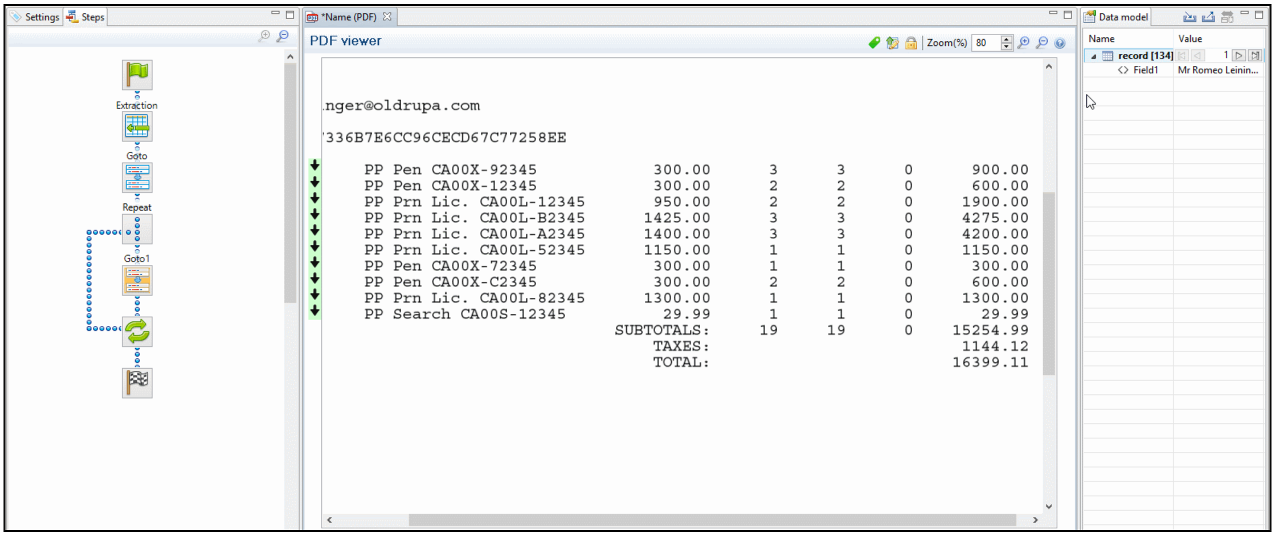
1. Select the text (SUBTOTALS for example) in the Viewer.
2. Right click on the selection and select **Add Repeat**.



```
r.leininger@oldrupa.com
EN
86EDFF7336B7E6CC96CECD67C77258EE

PPX027      PP Pen CA00X-92345         300.00     3     3     0     900.00
PPX019      PP Pen CA00X-12345         300.00     2     2     0     600.00
PPL006      PP Prn Lic. CA00L-12345    950.00     2     2     0    1900.00
PPL016      PP Prn Lic. CA00L-B2345   1425.00     3     3     0    4275.00
PPL015      PP Prn Lic. CA00L-A2345   1400.00     3     3     0    4200.00
PPL010      PP Prn Lic. CA00L-52345   1150.00     1     1     0    1150.00
PPX025      PP Pen CA00X-72345         300.00     1     1     0     300.00
PPX030      PP Pen CA00X-C2345         300.00     2     2     0     600.00
PPL013      PP Prn Lic. CA00L-82345   1300.00     1     1     0    1300.00
PPS004      PP Search CA00S-12345       29.99     1     1     0      29.99
                                   SUBTOTALS:    19    19     0   15254.99
                                       TAXES:                      1144.12
                                       TOTAL:                     16399.11
```

The **Repeat** step loops on all lines until the selected text is found. Since a record can extend on more than one page, lines that are not item lines must be excluded form the extraction. You can use any exclusive information like a "." or "," in prices or totals for example.

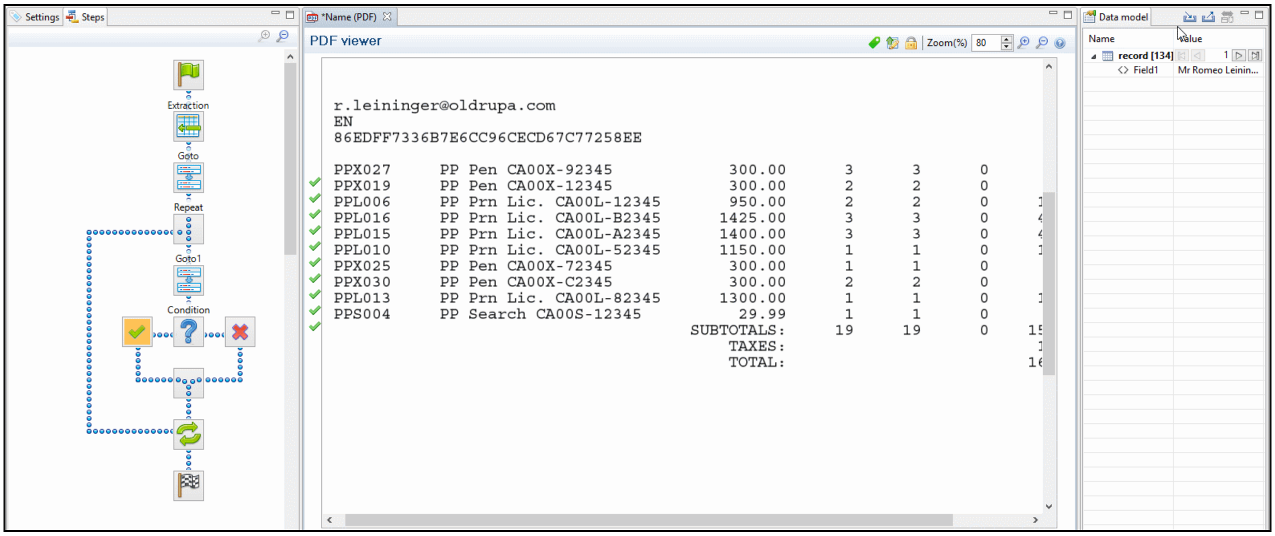
4. Select the "." or the "," in the total or in the price of the first line item.
5. Right click on the selected dot and select **Add Conditional**. In the Viewer pane, you will see a green check mark besides each included line or X for other lines.



```
.nger@oldrupa.com

'336B7E6CC96CECD67C77258EE

    PP Pen CA00X-92345         300.00     3     3     0     900.00
    PP Pen CA00X-12345         300.00     2     2     0     600.00
    PP Prn Lic. CA00L-12345    950.00     2     2     0    1900.00
    PP Prn Lic. CA00L-B2345   1425.00     3     3     0    4275.00
    PP Prn Lic. CA00L-A2345   1400.00     3     3     0    4200.00
    PP Prn Lic. CA00L-52345   1150.00     1     1     0    1150.00
    PP Pen CA00X-72345         300.00     1     1     0     300.00
    PP Pen CA00X-C2345         300.00     2     2     0     600.00
    PP Prn Lic. CA00L-82345   1300.00     1     1     0    1300.00
    PP Search CA00S-12345       29.99     1     1     0      29.99
                           SUBTOTALS:    19    19     0   15254.99
                               TAXES:                      1144.12
                               TOTAL:                     16399.11
```

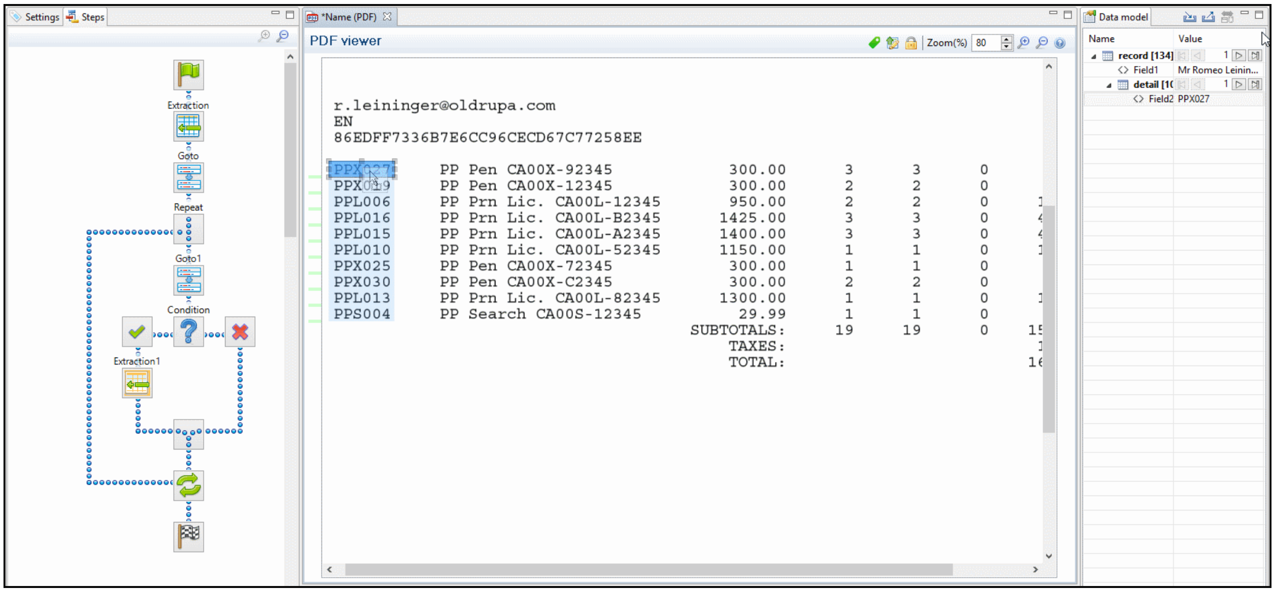
6. From the Viewer pane, select the first field on the left for the first line item.

7. Right click on the selection and select **Add Extraction**.



```
r.leininger@oldrupa.com
EN
86EDFF7336B7E6CC96CECD67C77258EE

PPX027     PP Pen CA00X-92345          300.00      3      3      0
PPX019     PP Pen CA00X-12345          300.00      2      2      0
PPL006     PP Prn Lic. CA00L-12345     950.00      2      2      0     1
PPL016     PP Prn Lic. CA00L-B2345    1425.00      3      3      0     4
PPL015     PP Prn Lic. CA00L-A2345    1400.00      3      3      0     4
PPL010     PP Prn Lic. CA00L-52345    1150.00      1      1      0     1
PPX025     PP Pen CA00X-72345          300.00      1      1      0
PPX030     PP Pen CA00X-C2345          300.00      2      2      0
PPL013     PP Prn Lic. CA00L-82345    1300.00      1      1      0     1
PPS004     PP Search CA00S-12345        29.99      1      1      0
                                   SUBTOTALS:     19     19      0    15
                                       TAXES:                            1
                                       TOTAL:                           16
```

8. Select the second field.
9. Right click on the selection and select **Add Extract Field**.



```
r.leininger@oldrupa.com
EN
86EDFF7336B7E6CC96CECD67C77258EE

PPX027     PP Pen CA00X-92345          300.00      3      3      0
PPX019     PP Pen CA00X-12345          300.00      2      2      0
PPL006     PP Prn Lic. CA00L-12345     950.00      2      2      0     1
PPL016     PP Prn Lic. CA00L-B2345    1425.00      3      3      0     4
PPL015     PP Prn Lic. CA00L-A2345    1400.00      3      3      0     4
PPL010     PP Prn Lic. CA00L-52345    1150.00      1      1      0     1
PPX025     PP Pen CA00X-72345          300.00      1      1      0
PPX030     PP Pen CA00X-C2345          300.00      2      2      0
PPL013     PP Prn Lic. CA00L-82345    1300.00      1      1      0     1
PPS004     PP Search CA00S-12345        29.99      1      1      0
                                   SUBTOTALS:     19     19      0    15
                                       TAXES:                            1
                                       TOTAL:                           16
```

10. Do the same for the rest of the fields along that same line item.

> **Note**
>
> The **Add Extract Field** step allows to add a field to an existing extract step while the **Add**

Extract Step creates a new step in the **Steps** pane. For optimization purposes, it is better to use **Add Extract Field** than to have a succession of **Add Extract** steps.

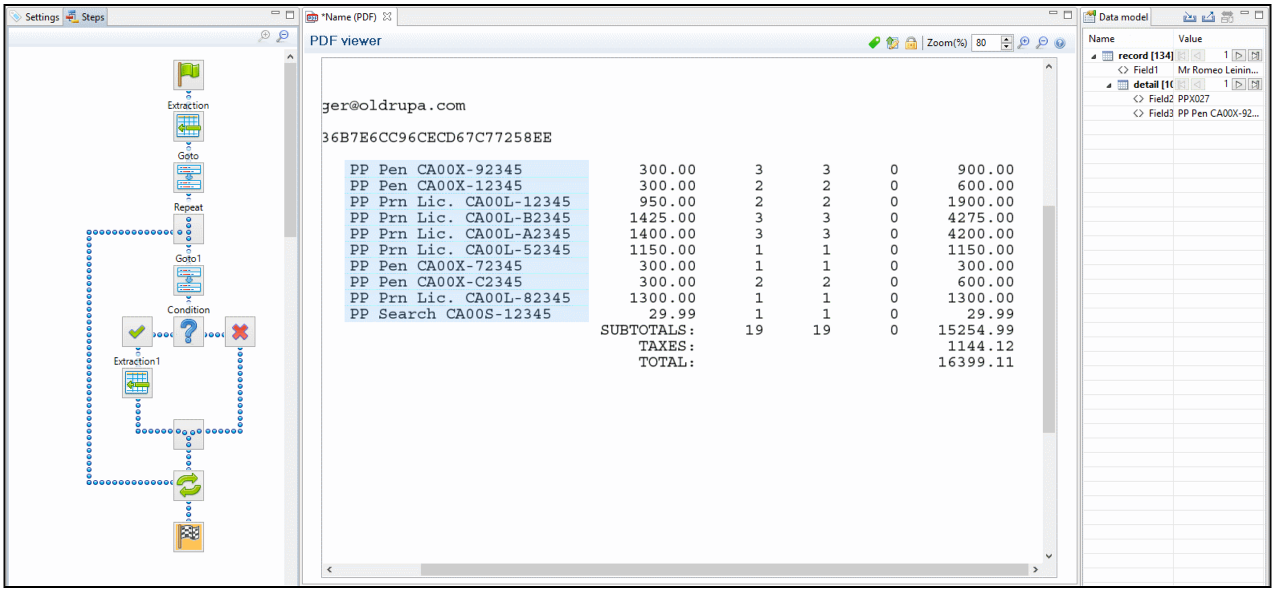
Note

You can also use the toolbar buttons instead of contextual menus, See Toolbars for more details.

Totals Information

After the loop step, the cursor position is at the end of line items. If the record contains sums or totals at the end of the line items list:

1. Select the amounts.
2. Click the **Repeat** step in the **Steps** panel.
3. Right click on the selection and select **Add a Step/Add Extraction**.



```
ger@oldrupa.com

36B7E6CC96CECD67C77258EE

  PP Pen CA00X-92345           300.00     3     3     0      900.00
  PP Pen CA00X-12345           300.00     2     2     0      600.00
  PP Prn Lic. CA00L-12345      950.00     2     2     0     1900.00
  PP Prn Lic. CA00L-B2345     1425.00     3     3     0     4275.00
  PP Prn Lic. CA00L-A2345     1400.00     3     3     0     4200.00
  PP Prn Lic. CA00L-52345     1150.00     1     1     0     1150.00
  PP Pen CA00X-72345           300.00     1     1     0      300.00
  PP Pen CA00X-C2345           300.00     2     2     0      600.00
  PP Prn Lic. CA00L-82345     1300.00     1     1     0     1300.00
  PP Search CA00S-12345         29.99     1     1     0       29.99
                          SUBTOTALS:     19    19     0    15254.99
                              TAXES:                         1144.12
                              TOTAL:                        16399.11
```

Name	Value
record [134]	1
Field1	Mr Romeo Leinin...
detail [10	1
Field2	PPX027
Field3	PP Pen CA00X-92...

Also Note That...

Note

- If no **Extract** step is currently selected in the **Steps** pane, a new **Extraction** step is created. If an **Extract** step is selected, new **Extract Fields** will be added to this existing step.
- Dragging a data selection or fields into a specific level of the Data Model (record or detail tables) will add the fields to that level.
- When dragging data into a detail table, the **Extract** step must be located within the appropriate **Repeat** step, otherwise the extract will not function properly.
- If a Data Model is loaded into the Data Model pane and if a field of the same name already exists in that level, dragging a named field into it (CSV, XML, Database) will extract the data to that field. Otherwise a new one with will be created.
- Dragging a single field onto a Data Model field will extract it to that field regardless of its name.
- **Action** steps could be added to the process described above in order to set the value for a Source Record property or execute a JavaScript code. For more information about the **Action** step, see The Action Step.

Note

To add an **Extract** step, you can also use JavaScript (select JavaScript from the **Mode** drop-down in the **Step Properties** pane). Please refer toJavaScript_in_DataMapper.htmfor more information.

Data Selection

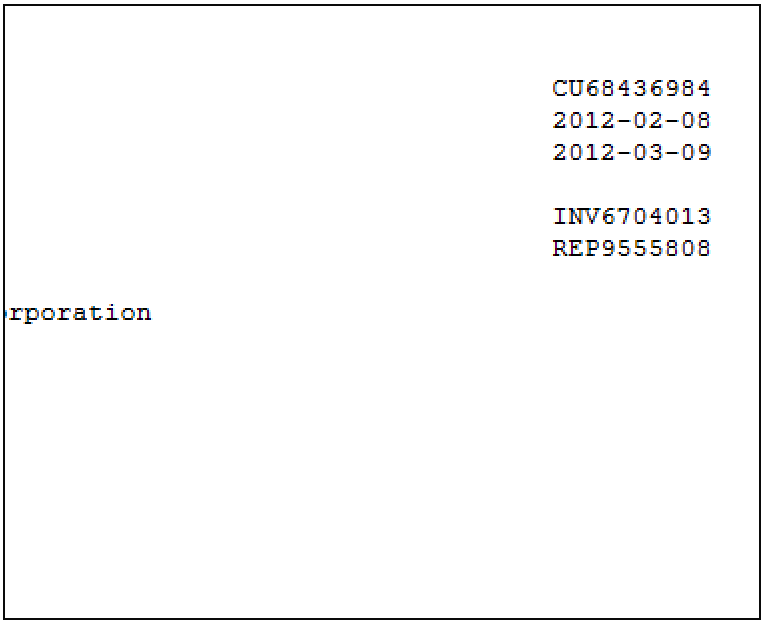
The following topics contain information about how to create and manipulate a data selection, and also how to create steps from it.

Selecting Data

From a Text File

The Text data viewer displays the text-based contents of the Data Sample that is currently active within the data mapping configuration in a grid-like fashion, with each character in the file being in a separate grid position.

To select data, click on a starting character within the grid, keeping the mouse button down, dragging to the end character and releasing the button. This creates a data selection that can contain multiple lines.

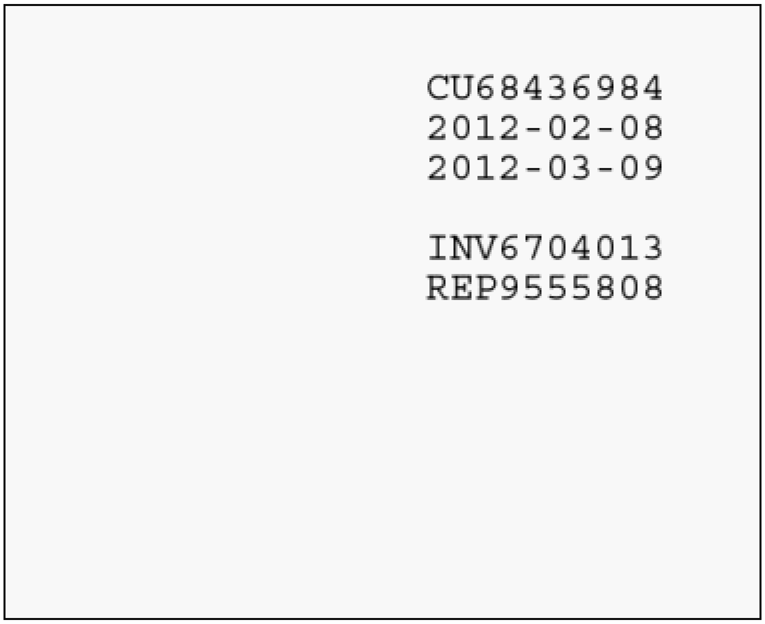


```
                                   CU68436984
                                   2012-02-08
                                   2012-03-09

                                   INV6704013
                                   REP9555808

rporation
```

From a PDF File

The PDF data viewer displays the PDF file contents of the Data Sample that is currently active within the data mapping configuration as pages.

To select data, click on a starting point, keeping the mouse button down, dragging to the end location and releasing the button. This creates a data selection that can contain multiple lines.

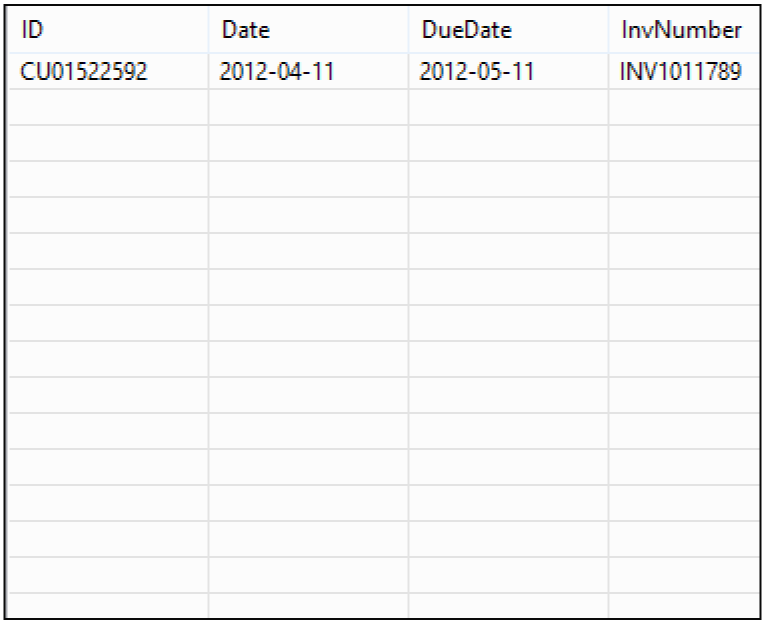


```
CU68436984
2012-02-08
2012-03-09

INV6704013
REP9555808
```

From a CSV or a Database File

The CSV/Database data viewer displays the field-based contents of the Data Sample that is currently active within the data mapping configuration in a grid-like fashion, with each field being in a separate grid position.

To select data, click on a starting point, keeping the mouse button down, dragging to the end location and releasing the button. This creates a data selection that can contain multiple lines.

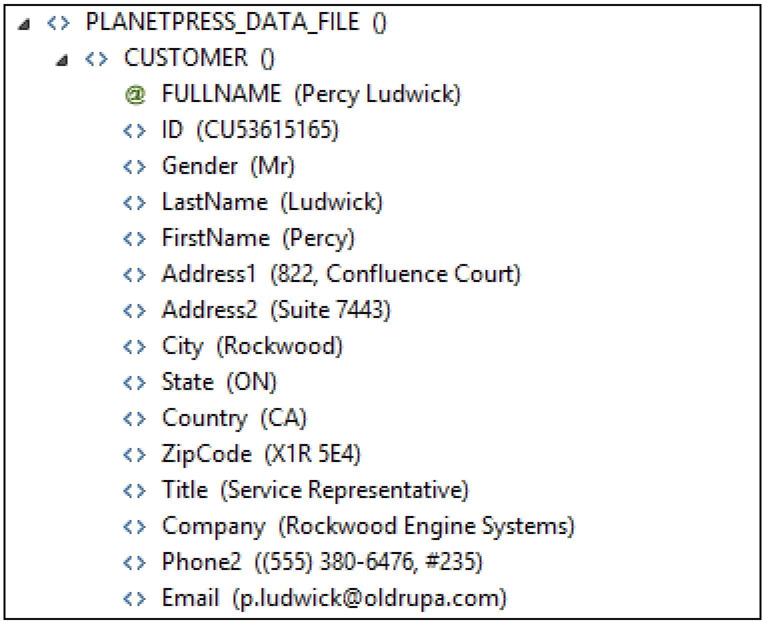


ID	Date	DueDate	InvNumber
CU01522592	2012-04-11	2012-05-11	INV1011789

From a XML File

The XML data viewer displays the XML-based contents of the Data Sample that is currently active within the data mapping configuration in a tree view, with XML fields displayed at each level of the Sample Record.

To select data, click on a starting point, keeping the mouse button down, dragging to the end location and releasing the button. This creates a data selection that can contain multiple lines.

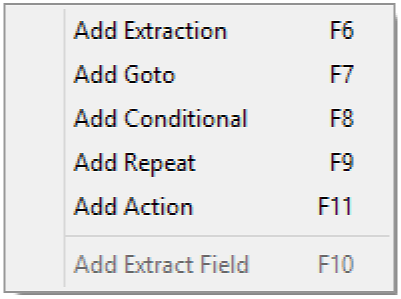


PLANETPRESS_DATA_FILE ()
CUSTOMER ()
FULLNAME (Percy Ludwick)
ID (CU53615165)
Gender (Mr)
LastName (Ludwick)
FirstName (Percy)
Address1 (822, Confluence Court)
Address2 (Suite 7443)
City (Rockwood)
State (ON)
Country (CA)
ZipCode (X1R 5E4)
Title (Service Representative)
Company (Rockwood Engine Systems)
Phone2 ((555) 380-6476, #235)
Email (p.ludwick@oldrupa.com)

Adding an Extraction Step from a Data Selection

To add a new step from the Data Viewer:

1. Create your data selection (refer to Creating a Data Selection for more information).
2. Right-clicking on the data selection and select the appropriate step.



See Type of Steps for more information on available steps.

Manipulating a Data Selection

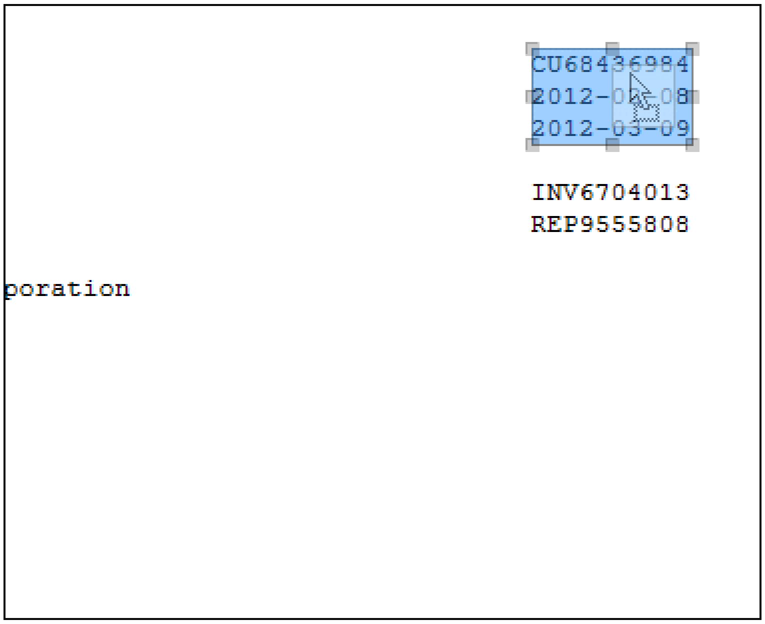
From a Text File

The Text data viewer displays the text-based contents of the Data Sample that is currently active within the data mapping configuration in a grid-like fashion, with each character in the file being in a separate grid position.

Once created, data selections can be modified and moved in order to change or extend the data included in the selection. You can also modify a data selection that is attached to a field extraction in an **Extract** step by double-clicking on the data selection in the Data Viewer and then modifying the data selection.

Moving a Data Selection

To move a data selection, click and hold anywhere on the data selection, move it to its new desired location and release the mouses.



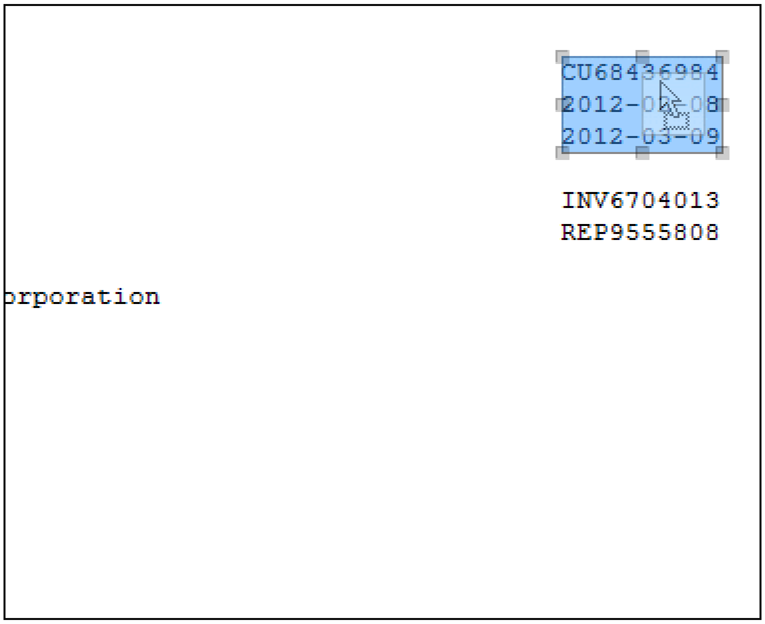
CU68436984
2012-03-08
2012-03-09

INV6704013
REP9555808

poration

Resizing a Data Selection

To resize a data selection, click and hold on one of the resize handles on the borders or corners, move them to the new size and release the mouse.



CU68436984
2012-02-08
2012-03-09

INV6704013
REP9555808

orporation

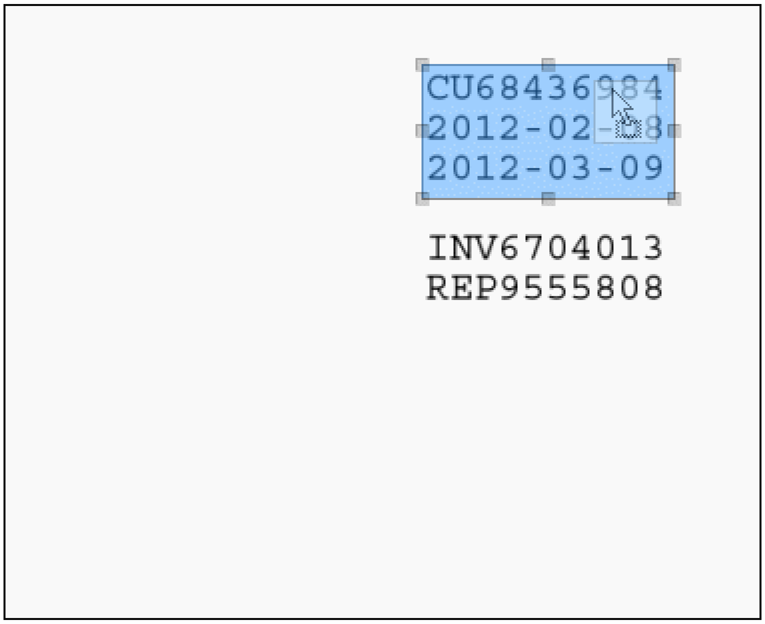
From a PDF File

The PDF data viewer displays the PDF file contents of the Data Sample that is currently active within the data mapping configuration as pages.

Once created, data selections can be modified and moved in order to change or extend the data included in the extraction. Moving a new data selection can be done directly if the data selection is new. Data selections that are attached to a field extraction in an **Extract** step can also be modified by double-clicking on the data selection in the Data Viewer and then modifying the data selection.

Moving a Data Selection

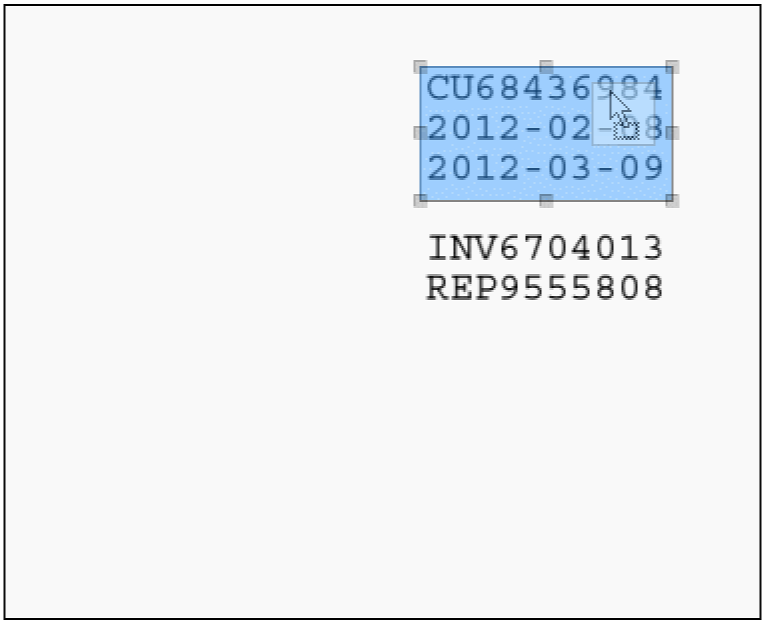
To move a data selection, click and hold anywhere on the data selection, move it to its new desired location and release the mouse.


CU68436984
2012-02-08
2012-03-09

INV6704013
REP9555808

Resizing a Data Selection

To resize a data selection, click and hold on one of the resize handles on the borders or corners, move them to the new size and release the mouse.


CU68436984
2012-02-08
2012-03-09

INV6704013
REP9555808

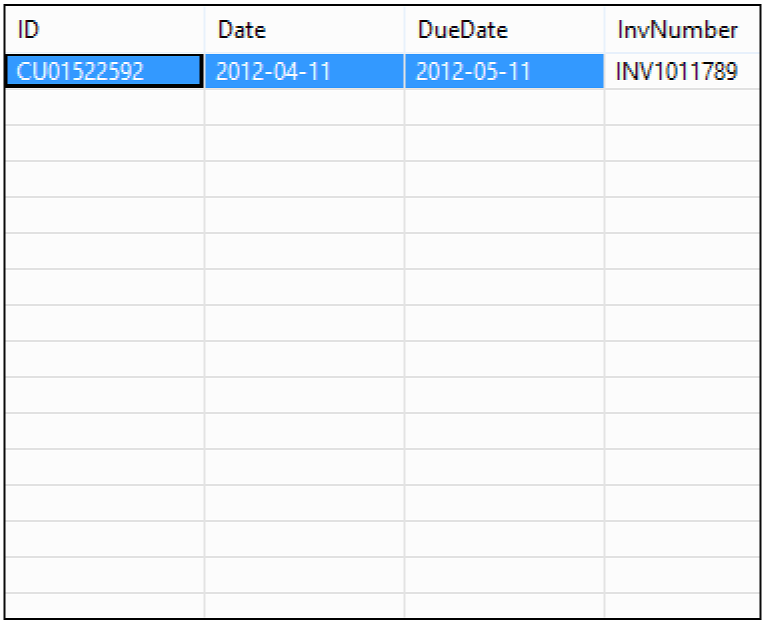
From a CSV or a Database File

The CSV/Database data viewer displays the field-based contents of the Data Sample that is currently active within the data mapping configuration in a grid-like fashion, with each field being in a separate grid position.

Once created, data selections can be modified. Modifying a new data selection can be done directly if the data selection is new.

Modifying a Data Selection

To modify a data selection, press and hold the **CTRL** key click, and click on a field to add it or remove it from the selection (the blue color indicates that the field is part of the current selection).



ID	Date	DueDate	InvNumber
CU01522592	2012-04-11	2012-05-11	INV1011789

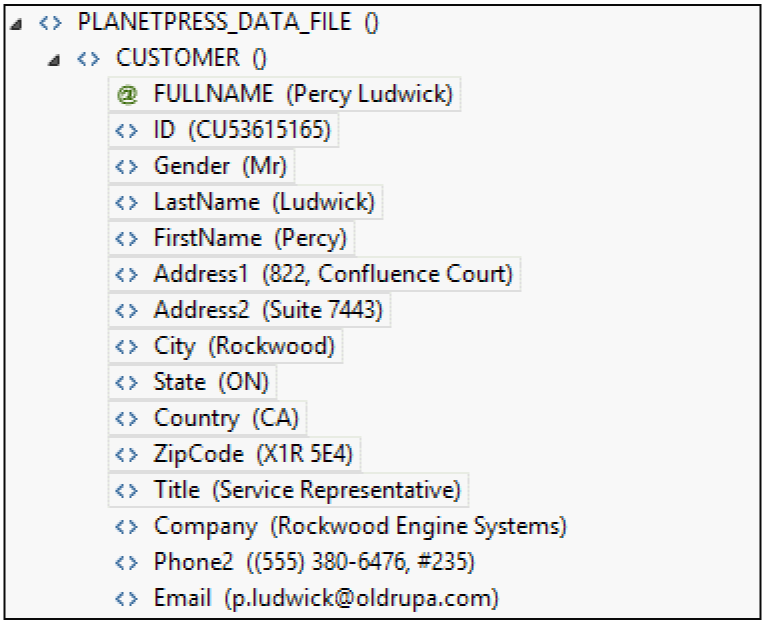
From a XML File

The XML data viewer displays the XML-based contents of the Data Sample that is currently active within the data mapping configuration in a tree view, with XML fields displayed at each level of the Sample Record.

Once created, data selections can be modified. Modifying a new data selection can be done directly if the data selection is new.

Modifying a Data Selection

To modify a data selection, press and hold the **CTRL** key click, and click on a field to add it or remove it from the selection (the blue color indicates that the field is part of the current selection).



```
◢ <> PLANETPRESS_DATA_FILE ()
    ◢ <> CUSTOMER ()
          @ FULLNAME (Percy Ludwick)
          <> ID (CU53615165)
          <> Gender (Mr)
          <> LastName (Ludwick)
          <> FirstName (Percy)
          <> Address1 (822, Confluence Court)
          <> Address2 (Suite 7443)
          <> City (Rockwood)
          <> State (ON)
          <> Country (CA)
          <> ZipCode (X1R 5E4)
          <> Title (Service Representative)
          <> Company (Rockwood Engine Systems)
          <> Phone2 ((555) 380-6476, #235)
          <> Email (p.ludwick@oldrupa.com)
```

Steps

Steps are executed sequentially, from top to bottom. Inside conditions, some steps may be skipped altogether when they are on a particular branch, whereas in loops several steps may be repeated a number of times. The Preprocessor and Postprocessor steps are special in that the former can be used to modify the incoming data prior to executing the rest of the workflow while the latter can be used to further process the Data Set after the entire workflow has been executed.

You will find below a list of different steps that can be added to a Data Mapping Workflow.

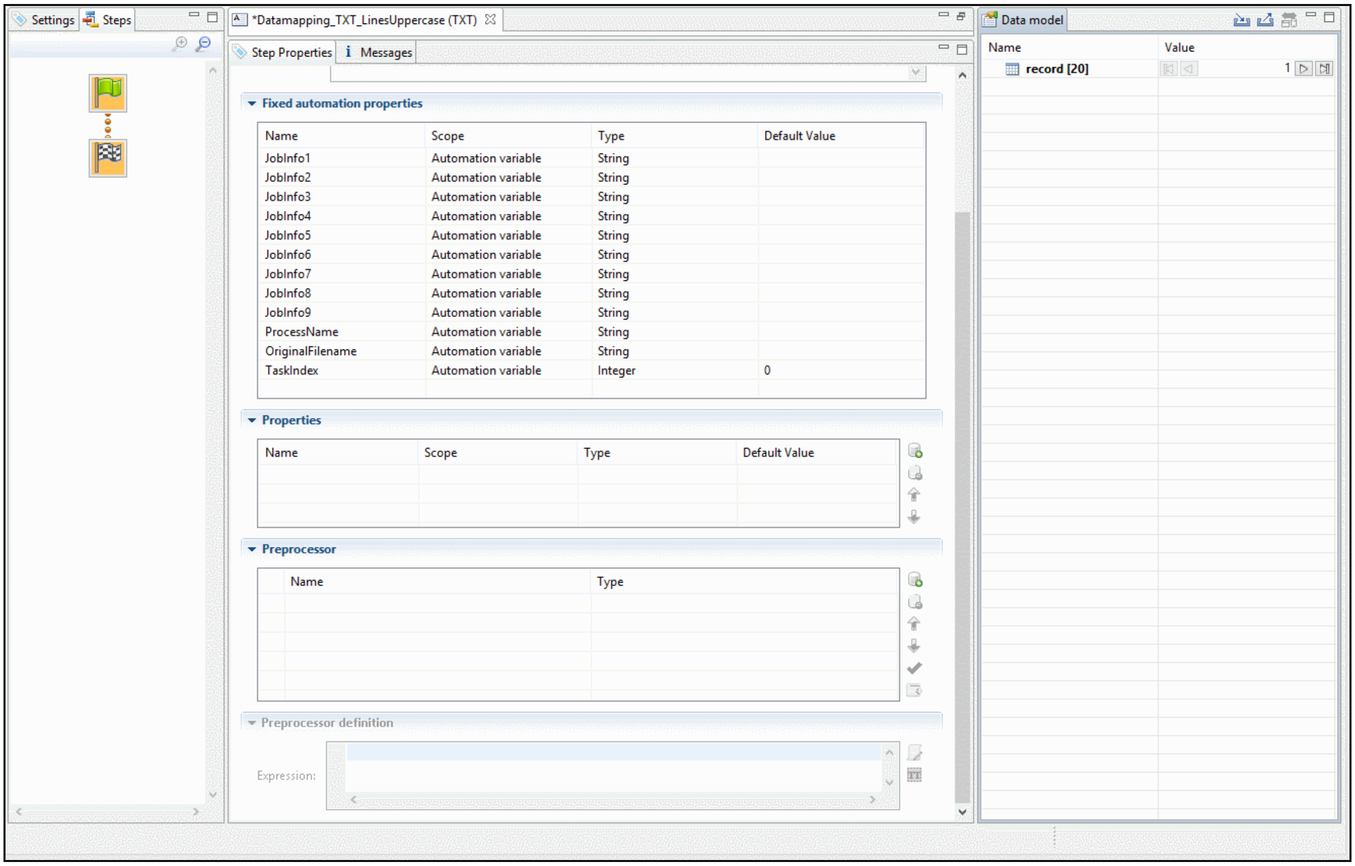
> **Note**
>
> For more details about operations that can be performed on steps, please refer to The Steps Pane Interface.

Preprocessor and Postprocessor

Preprocessors

Data preprocessors allow the application to perform actions on the data file itself before it is handed over to the Data Mapping workflow

Some preprocessors might for instance, using a JavaScript expression, convert all characters of a text file from lowercase to uppercase prior to running Data Mapping workflow :



Fixed automation properties

Name	Scope	Type	Default Value
JobInfo1	Automation variable	String	
JobInfo2	Automation variable	String	
JobInfo3	Automation variable	String	
JobInfo4	Automation variable	String	
JobInfo5	Automation variable	String	
JobInfo6	Automation variable	String	
JobInfo7	Automation variable	String	
JobInfo8	Automation variable	String	
JobInfo9	Automation variable	String	
ProcessName	Automation variable	String	
OriginalFilename	Automation variable	String	
TaskIndex	Automation variable	Integer	0

Properties

Name	Scope	Type	Default Value

Preprocessor

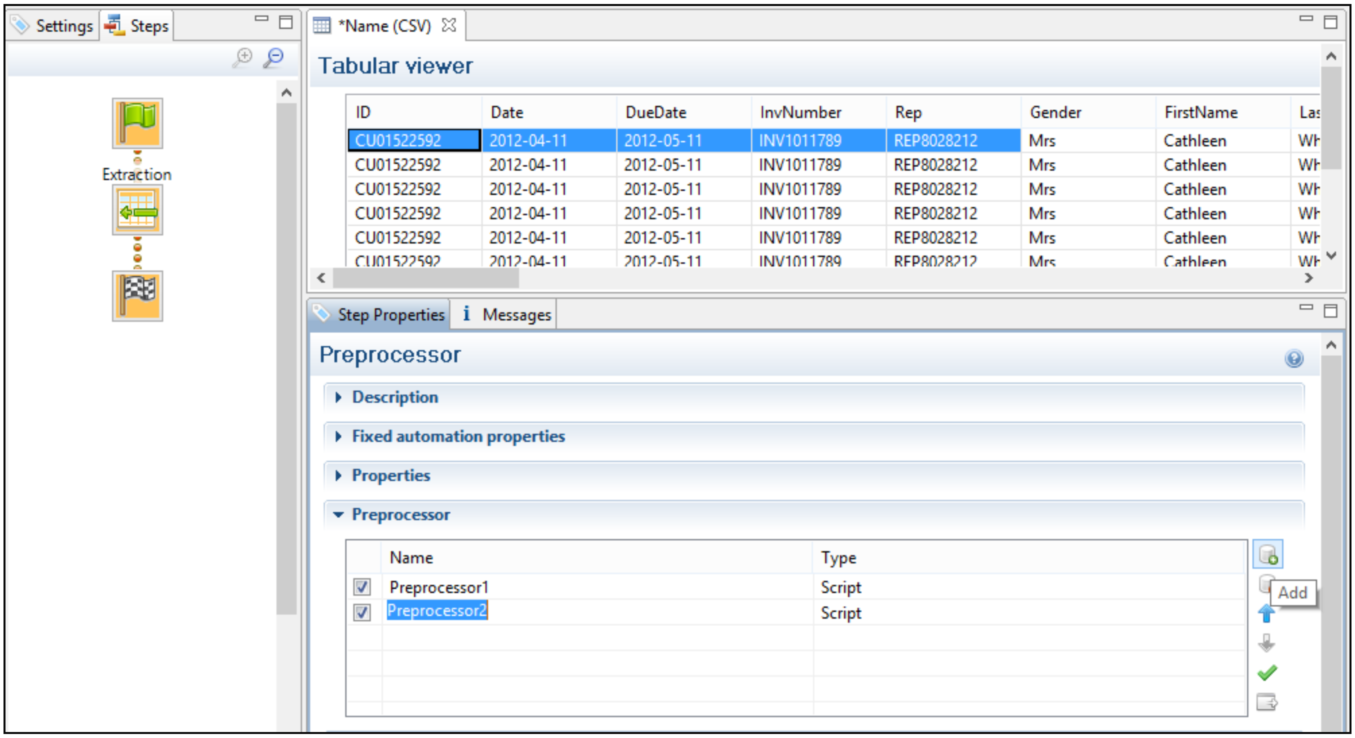
Name	Type

Preprocessor definition

Expression:

Name	Value
record [20]	1

Any number of preprocessing tasks can be added to a workflow and run in sequence before the data is sent to the Data Mapping workflow. Click the button to add a Preprocessor to the list.



Tabular viewer

ID	Date	DueDate	InvNumber	Rep	Gender	FirstName	Las
CU01522592	2012-04-11	2012-05-11	INV1011789	REP8028212	Mrs	Cathleen	Wh
CU01522592	2012-04-11	2012-05-11	INV1011789	REP8028212	Mrs	Cathleen	Wh
CU01522592	2012-04-11	2012-05-11	INV1011789	REP8028212	Mrs	Cathleen	Wh
CU01522592	2012-04-11	2012-05-11	INV1011789	REP8028212	Mrs	Cathleen	Wh
CU01522592	2012-04-11	2012-05-11	INV1011789	REP8028212	Mrs	Cathleen	Wh
CU01522592	2012-04-11	2012-05-11	INV1011789	REP8028212	Mrs	Cathleen	Wh

Preprocessor

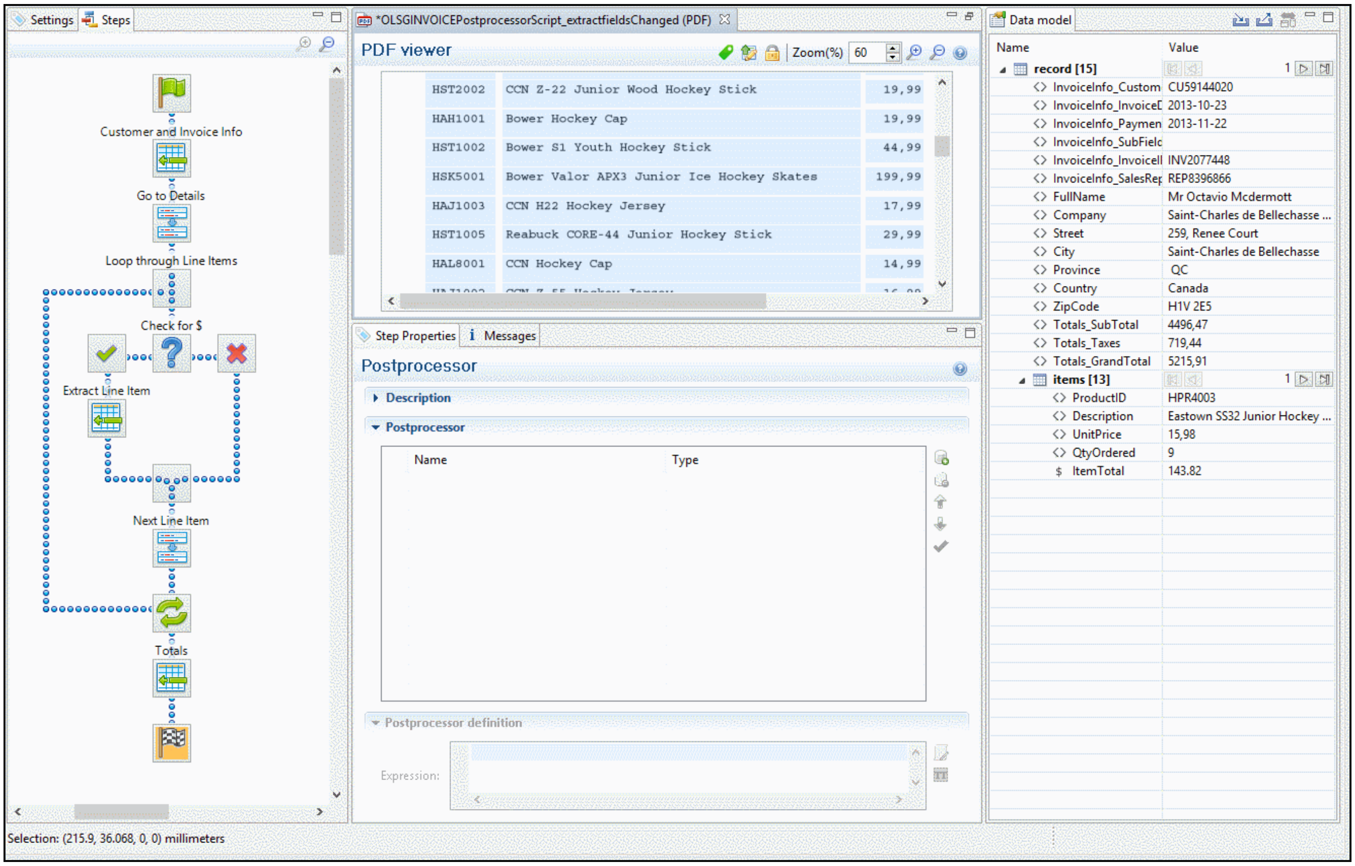
- Description
- Fixed automation properties
- Properties
- Preprocessor

	Name	Type
☑	Preprocessor1	Script
☑	Preprocessor2	Script

Postprocessors

Data postprocessors allow the application to extract data that was stored in the data model once the workflow is complete. For example, the post processor can export all data or only addresses, to a CSV file which can then be given directly to a postal addresses verification application.

Some postprocessors might for instance, using a JavaScript expression, create a CSV file from the Record set:



HST2002	CCN Z-22 Junior Wood Hockey Stick	19,99
HAH1001	Bower Hockey Cap	19,99
HST1002	Bower S1 Youth Hockey Stick	44,99
HSK5001	Bower Valor APX3 Junior Ice Hockey Skates	199,99
HAJ1003	CCN H22 Hockey Jersey	17,99
HST1005	Reabuck CORE-44 Junior Hockey Stick	29,99
HAL8001	CCN Hockey Cap	14,99

Postprocessor

- Description
- Postprocessor

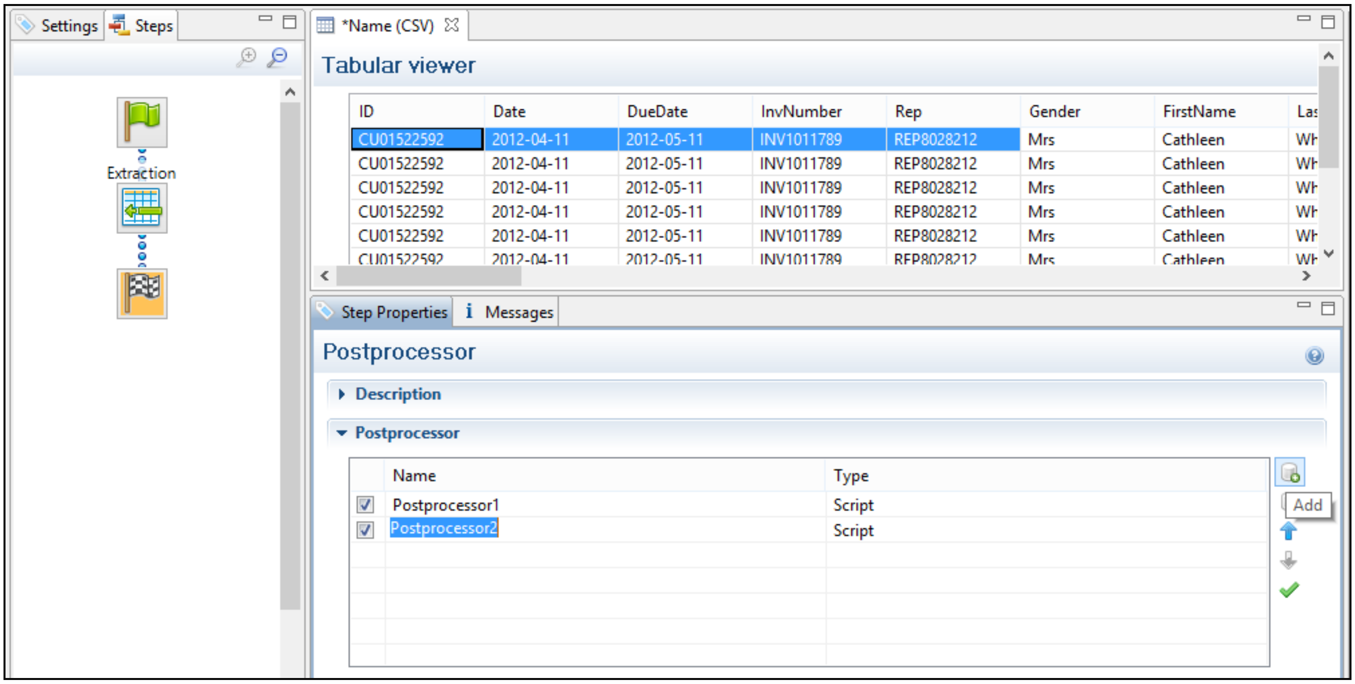
Name	Type

- Postprocessor definition

Expression:

Name	Value
record [15]	1
InvoiceInfo_Custom	CU59144020
InvoiceInfo_InvoiceD	2013-10-23
InvoiceInfo_Paymen	2013-11-22
InvoiceInfo_SubFielc	
InvoiceInfo_InvoiceI	INV2077448
InvoiceInfo_SalesRep	REP8396866
FullName	Mr Octavio Mcdermott
Company	Saint-Charles de Bellechasse ...
Street	259, Renee Court
City	Saint-Charles de Bellechasse
Province	QC
Country	Canada
ZipCode	H1V 2E5
Totals_SubTotal	4496,47
Totals_Taxes	719,44
Totals_GrandTotal	5215,91
items [13]	1
ProductID	HPR4003
Description	Eastown SS32 Junior Hockey ...
UnitPrice	15,98
QtyOrdered	9
$ ItemTotal	143.82

Any number of postprocessing tasks can be added to a workflow and be executed after the Data Mapping workflow is complete. Click the ⊕ button to add a Postprocessor to the list.



Tabular viewer

ID	Date	DueDate	InvNumber	Rep	Gender	FirstName	Las
CU01522592	2012-04-11	2012-05-11	INV1011789	REP8028212	Mrs	Cathleen	Wh
CU01522592	2012-04-11	2012-05-11	INV1011789	REP8028212	Mrs	Cathleen	Wh
CU01522592	2012-04-11	2012-05-11	INV1011789	REP8028212	Mrs	Cathleen	Wh
CU01522592	2012-04-11	2012-05-11	INV1011789	REP8028212	Mrs	Cathleen	Wh
CU01522592	2012-04-11	2012-05-11	INV1011789	REP8028212	Mrs	Cathleen	Wh
CU01522592	2012-04-11	2012-05-11	INV1011789	REP8028212	Mrs	Cathleen	Wh

Postprocessor

Description

Postprocessor

	Name	Type
☑	Postprocessor1	Script
☑	Postprocessor2	Script

Extract

The **Extract** step is the heart of the DataMapper software. It takes information from the Data Sample and places it in the Extracted Record within the Record Set.

Please refer to Data Extraction for an animated example.

For more information on how to add a step, please refer to Toolbar, Menus or Shortcut Keys under the **Interface** section.

> **Note**
>
> To add an **Extract** step, you can also use JavaScript (select JavaScript from the **Mode** drop-down in the **Step Properties** pane). Please refer toJavaScript_in_DataMapper.htmfor more information.

Properties

You can also further customize the step properties. Please refer to The Step Properties Interface for more information.

Goto

The cursor position determines how data is extracted from the Source Record. The cursor in a Source Record is always at a specific position.

In a Source Record, the cursor position starts off at the top-left corner of the Sample Data. When the **Goto** step is used, that cursor position is moved to the location (either relative or absolute) set by the **Goto** step. In the case of a **Goto** step within a **Repeat** step, the cursor position will gradually be moved with each loop of the repeat step, for example when extracting transactional data.

Since data extracted by the **Extract** step is always relative to the current cursor position, this becomes useful when extracting data at the end of the loop. For instance, when a "Totals" line appears at the end of line items in an invoice, the **Extract** step can find the appropriate position of the "Totals" simply by looking at a certain distance (or offset) from the last line item and always find the "Totals" at the same place.

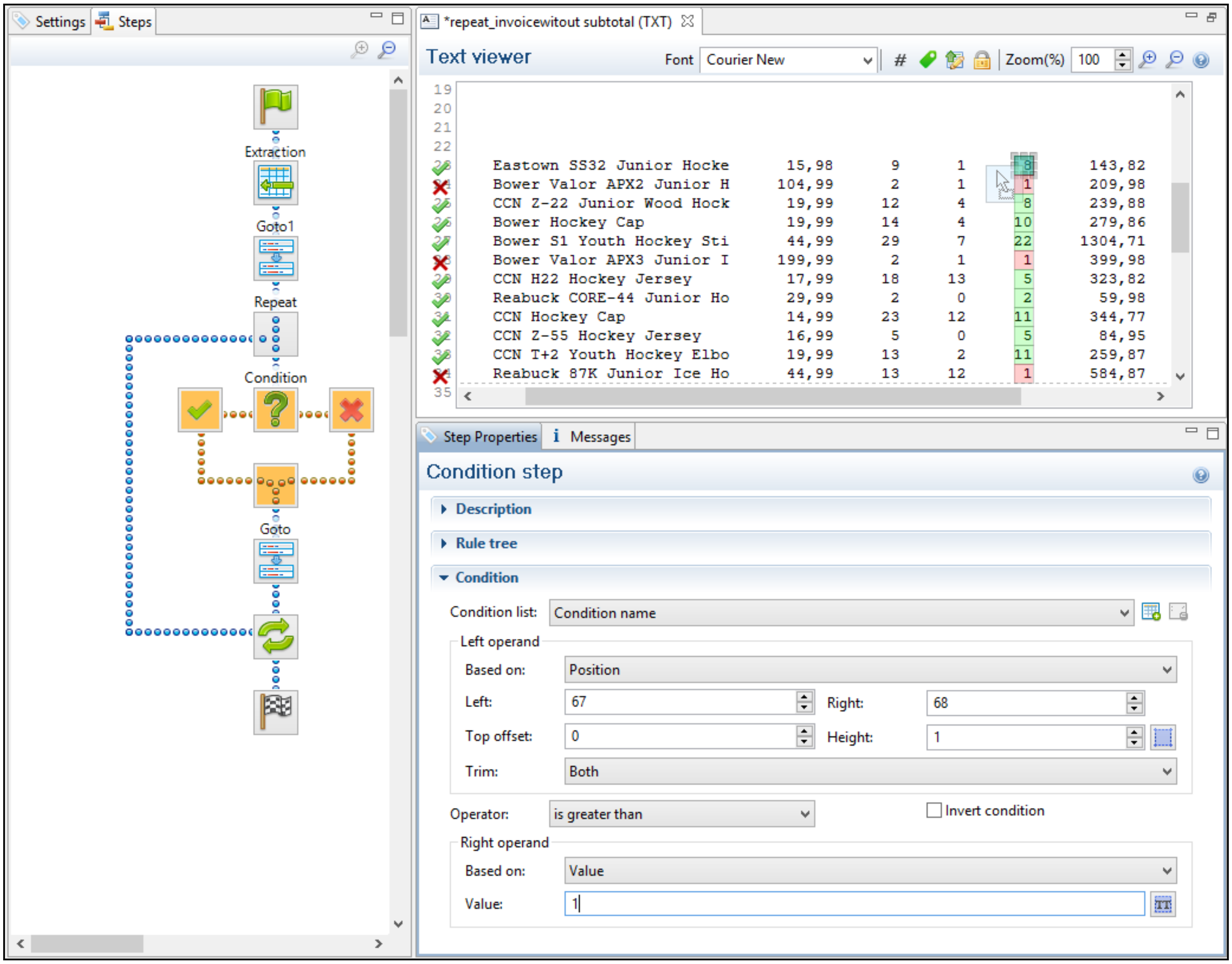
For more information on how to add a step, please refer to Toolbar, Menus or Shortcut Keys under the **Interface** section.

Properties

You can also further customize the step properties. Please refer to The Step Properties Interface for more information.

Condition

A **Condition** step splits the extraction workflow into two separate branches, one that is executed when the condition is **True**, the other when it is **False**. In the Viewer on the left side of the window, an icon is displayed indicating the result of the evaluation: ✓ when true and ✗ when false.



Font: Courier New   Zoom(%): 100

```
 Eastown SS32 Junior Hocke     15,98     9      1      8     143,82
 Bower Valor APX2 Junior H    104,99     2      1      1     209,98
 CCN Z-22 Junior Wood Hock     19,99    12      4      8     239,88
 Bower Hockey Cap              19,99    14      4     10     279,86
 Bower S1 Youth Hockey Sti     44,99    29      7     22    1304,71
 Bower Valor APX3 Junior I    199,99     2      1      1     399,98
 CCN H22 Hockey Jersey         17,99    18     13      5     323,82
 Reabuck CORE-44 Junior Ho     29,99     2      0      2      59,98
 CCN Hockey Cap                14,99    23     12     11     344,77
 CCN Z-55 Hockey Jersey        16,99     5      0      5      84,95
 CCN T+2 Youth Hockey Elbo     19,99    13      2     11     259,87
 Reabuck 87K Junior Ice Ho     44,99    13     12      1     584,87
```

Condition step

Condition

Condition list: Condition name

Left operand

Based on: Position

Left: 67   Right: 68

Top offset: 0   Height: 1

Trim: Both

Operator: is greater than   □ Invert condition

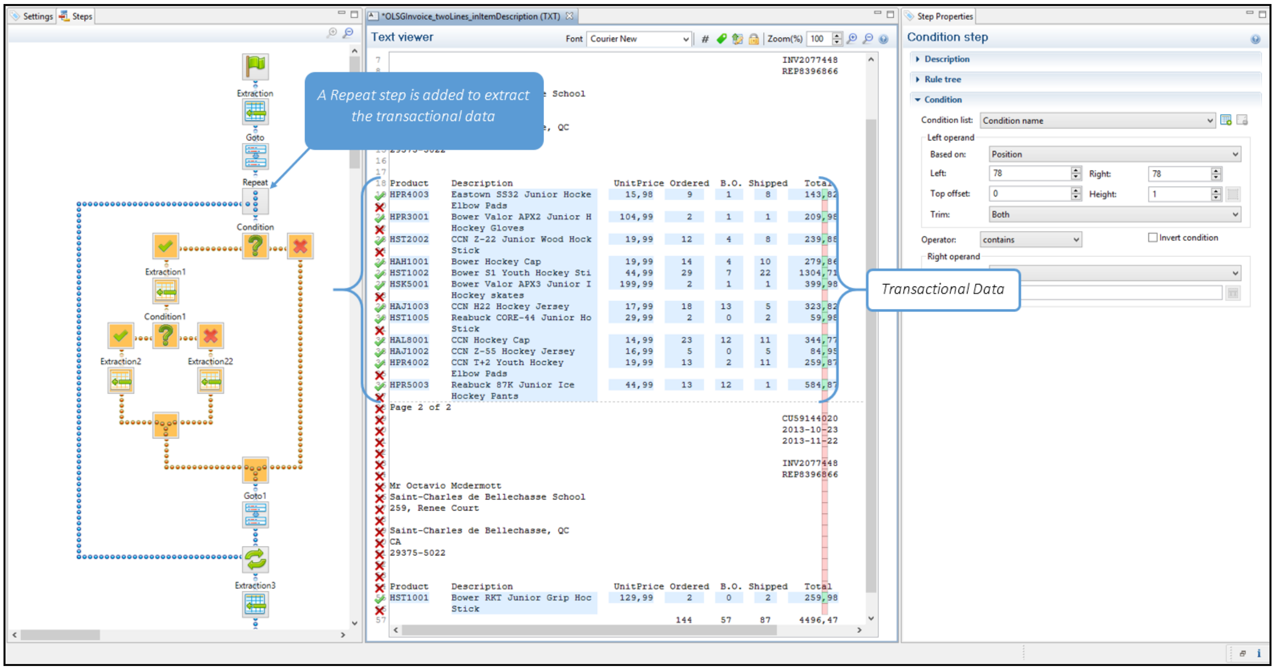
Right operand

Based on: Value

Value: 1

A **Condition** step is used when the data extraction must be made based on specific criteria. In the example shown in the following figure, the transactional data must be extracted according to two main criteria. First, the line item must include an amount of money and secondly, the lines that include a Description field on two lines have to be extracted as a single record.

1. Since the extraction is for transactional data, a **Repeat** step is first added.



A Repeat step is added to extract the transactional data

Transactional Data

```
                                                                          INV2077448
                                                                          REP8396866
                                              School

                                             , QC

Product      Description              UnitPrice Ordered  B.O. Shipped    Total
HPR4003      Eastown SS32 Junior Hocke     15,98      9     1      8    143,82
             Elbow Pads
HPR3001      Bower Valor APX2 Junior H    104,99      2     1      1    209,98
             Hockey Gloves
HST2002      CCN Z-22 Junior Wood Hock     19,99     12     4      8    239,88
             Stick
HAH1001      Bower Hockey Cap              19,99     14     4     10    279,86
HST1002      Bower S1 Youth Hockey Sti     44,99     29     7     22   1304,71
HSK5001      Bower Valor APX3 Junior I    199,99      2     1      1    399,98
             Hockey skates
HAJ1003      CCN H22 Hockey Jersey         17,99     18    13      5    323,82
HST1005      Reabuck CORE-44 Junior Ho     29,99      2     0      2     59,98
             Stick
HAL8001      CCN Hockey Cap                14,99     23    12     11    344,77
HAJ1002      CCN Z-55 Hockey Jersey        16,99      5     0      5     84,95
HPR4002      CCN T+2 Youth Hockey          19,99     13     2     11    259,87
             Elbow Pads
HPR5003      Reabuck 87K Junior Ice        44,99     13    12      1    584,87
             Hockey Pants
Page 2 of 2
                                                                          CU59144020
                                                                          2013-10-23
                                                                          2013-11-22

                                                                          INV2077448
                                                                          REP8396866
Mr Octavio Mcdermott
Saint-Charles de Bellechasse School
259, Renee Court

Saint-Charles de Bellechasse, QC
CA
29375-5022


Product      Description              UnitPrice Ordered  B.O. Shipped    Total
HST1001      Bower RKT Junior Grip Hoc    129,99      2     0      2    259,98
             Stick
                                                    144    57     87   4496,47
```

Condition step

Condition list: Condition name

Left operand

Based on: Position

Left: 78 Right: 78

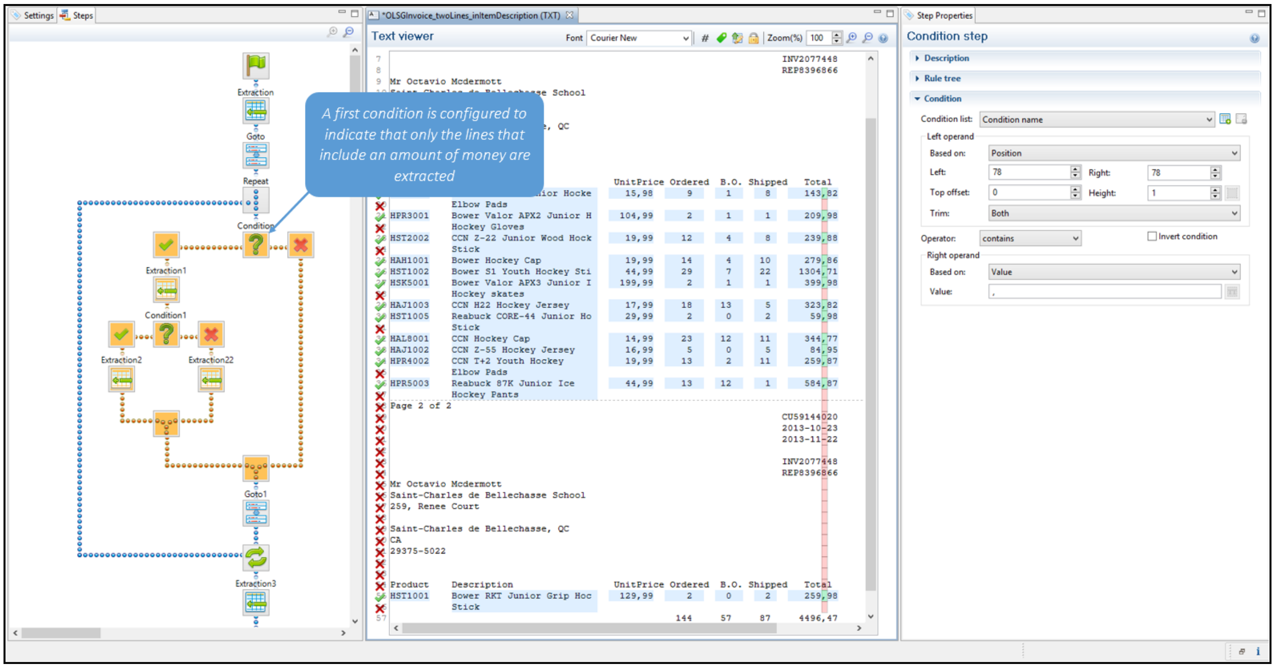
Top offset: 0 Height: 1

Trim: Both

Operator: contains ☐ Invert condition

Right operand

2. A first condition is added to determine whether the line should be considered for the extraction.



A first condition is configured to indicate that only the lines that include an amount of money are extracted

```
                                                                          INV2077448
                                                                          REP8396866
Mr Octavio Mcdermott
                                              School

                                             , QC

                                      UnitPrice Ordered  B.O. Shipped    Total
                                   ior Hocke   15,98      9     1      8    143,82
             Elbow Pads
HPR3001      Bower Valor APX2 Junior H    104,99      2     1      1    209,98
             Hockey Gloves
HST2002      CCN Z-22 Junior Wood Hock     19,99     12     4      8    239,88
             Stick
HAH1001      Bower Hockey Cap              19,99     14     4     10    279,86
HST1002      Bower S1 Youth Hockey Sti     44,99     29     7     22   1304,71
HSK5001      Bower Valor APX3 Junior I    199,99      2     1      1    399,98
             Hockey skates
HAJ1003      CCN H22 Hockey Jersey         17,99     18    13      5    323,82
HST1005      Reabuck CORE-44 Junior Ho     29,99      2     0      2     59,98
             Stick
HAL8001      CCN Hockey Cap                14,99     23    12     11    344,77
HAJ1002      CCN Z-55 Hockey Jersey        16,99      5     0      5     84,95
HPR4002      CCN T+2 Youth Hockey          19,99     13     2     11    259,87
             Elbow Pads
HPR5003      Reabuck 87K Junior Ice        44,99     13    12      1    584,87
             Hockey Pants
Page 2 of 2
                                                                          CU59144020
                                                                          2013-10-23
                                                                          2013-11-22

                                                                          INV2077448
                                                                          REP8396866
Mr Octavio Mcdermott
Saint-Charles de Bellechasse School
259, Renee Court

Saint-Charles de Bellechasse, QC
CA
29375-5022


Product      Description              UnitPrice Ordered  B.O. Shipped    Total
HST1001      Bower RKT Junior Grip Hoc    129,99      2     0      2    259,98
             Stick
                                                    144    57     87   4496,47
```

Condition step

Condition list: Condition name

Left operand

Based on: Position

Left: 78 Right: 78

Top offset: 0 Height: 1

Trim: Both

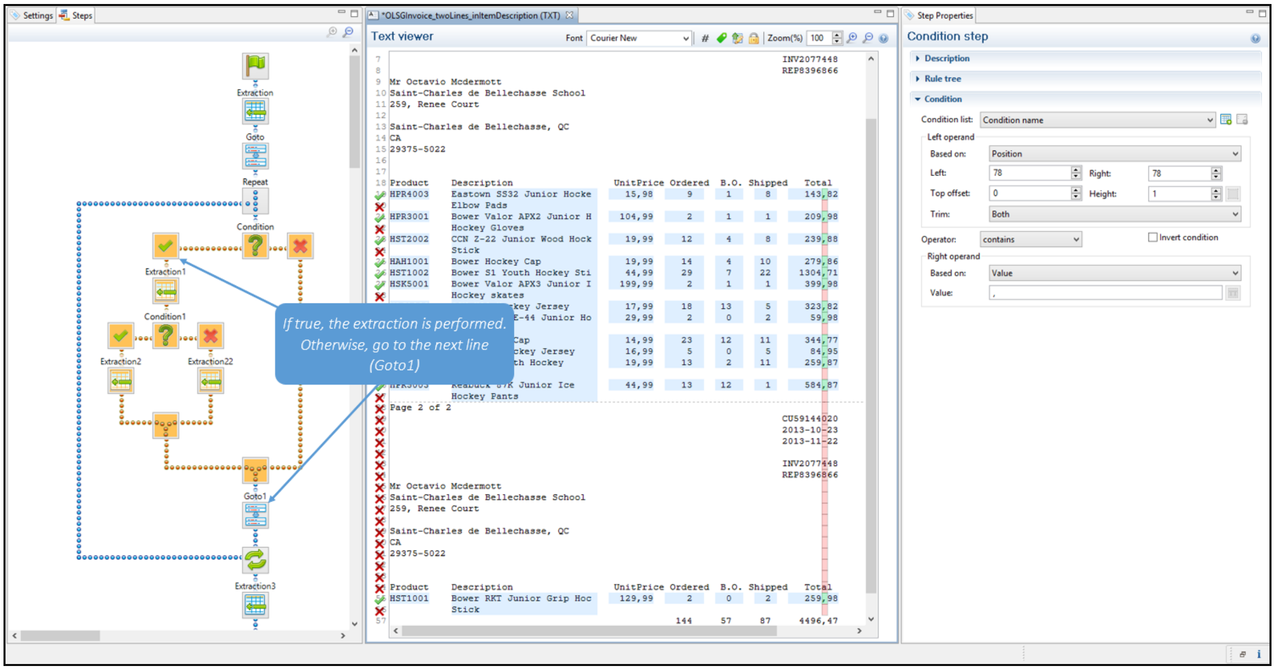
Operator: contains ☐ Invert condition

Right operand

Based on: Value

Value: ,

3. The extraction is performed if the condition is true.



If true, the extraction is performed. Otherwise, go to the next line (Goto1)

Condition step

- Description
- Rule tree
- Condition

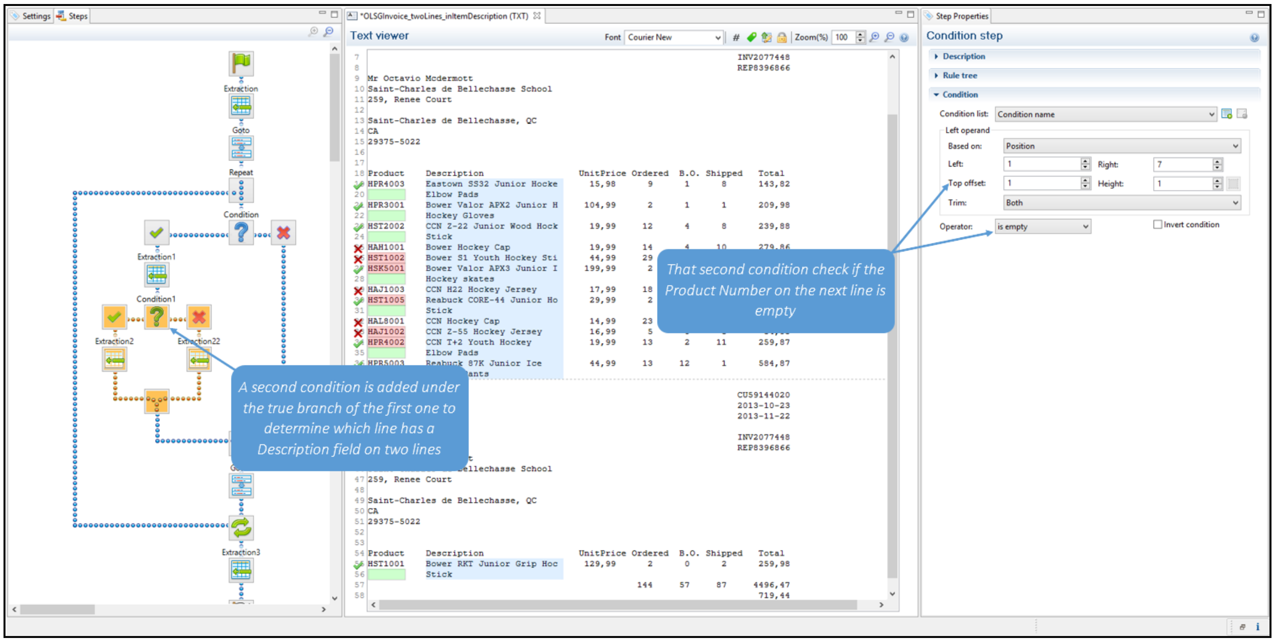
Condition list: Condition name

Left operand
Based on: Position
Left: 78 Right: 78
Top offset: 0 Height: 1
Trim: Both

Operator: contains ☐ Invert condition

Right operand
Based on: Value
Value: ,

4. Under the true branch of the first condition, a second condition is added for Description fields on two lines.



That second condition check if the Product Number on the next line is empty

A second condition is added under the true branch of the first one to determine which line has a Description field on two lines

Condition step

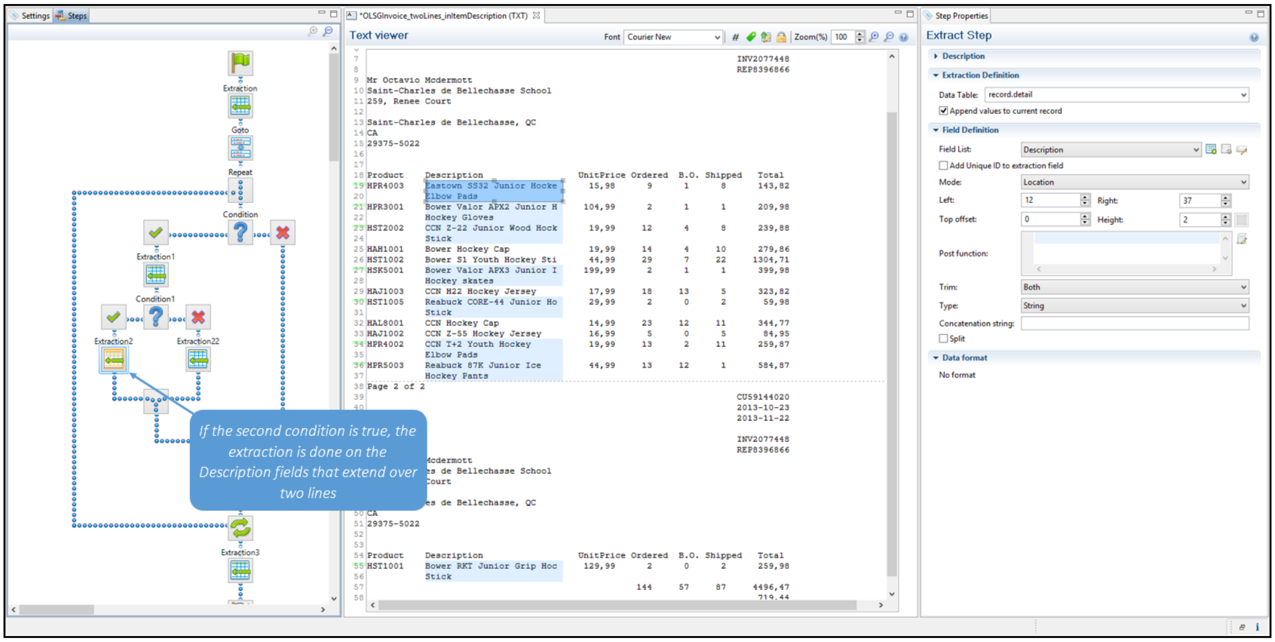
- Description
- Rule tree
- Condition

Condition list: Condition name

Left operand
Based on: Position
Left: 1 Right: 7
Top offset: 1 Height: 1
Trim: Both

Operator: is empty ☐ Invert condition

5. The extraction performed under the true branch of the second condition.



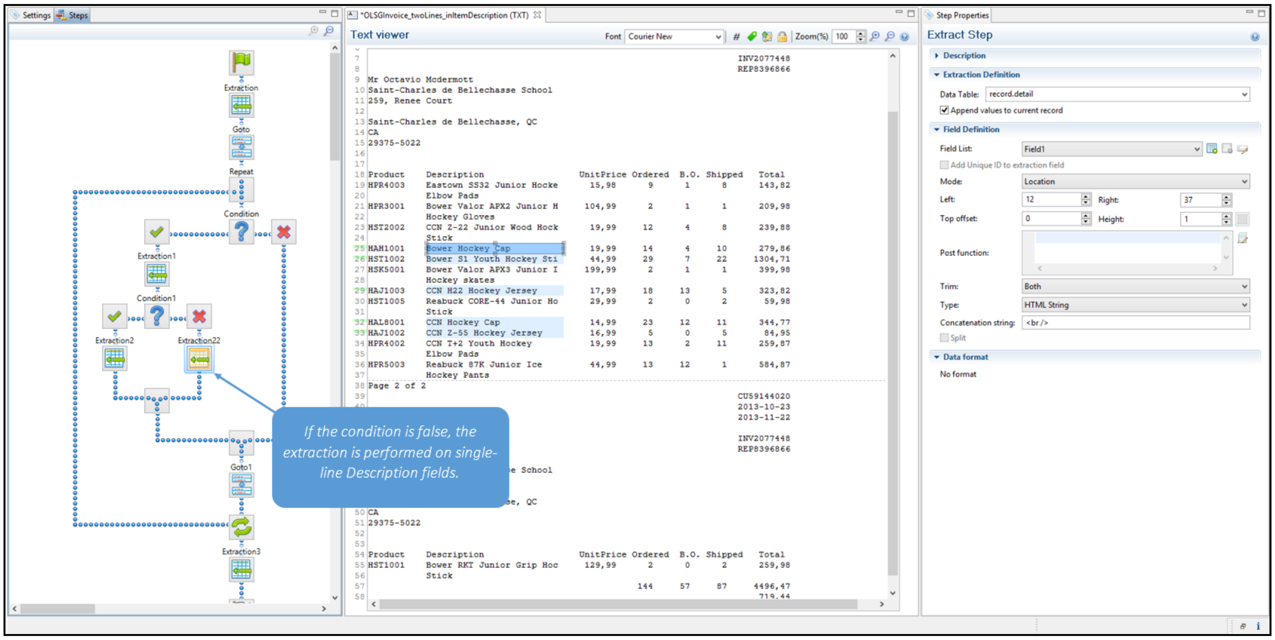
If the second condition is true, the extraction is done on the Description fields that extend over two lines

```
                                                                           INV2077448
                                                                           REP8396866
Mr Octavio Mcdermott
Saint-Charles de Bellechasse School
259, Renee Court

Saint-Charles de Bellechasse, QC
CA
29375-5022


Product     Description                  UnitPrice Ordered  B.O. Shipped   Total
HPR4003     Eastown SS32 Junior Hocke      15,98       9      1      8     143,82
            Elbow Pads
HPR3001     Bower Valor APX2 Junior H     104,99       2      1      1     209,98
            Hockey Gloves
HST2002     CCN Z-22 Junior Wood Hock      19,99      12      4      8     239,88
            Stick
HAH1001     Bower Hockey Cap               19,99      14      4     10     279,86
HST1002     Bower S1 Youth Hockey Sti      44,99      29      7     22    1304,71
HSK5001     Bower Valor APX3 Junior I     199,99       2      1      1     399,98
            Hockey skates
HAJ1003     CCN H22 Hockey Jersey          17,99      18     13      5     323,82
HST1005     Reabuck CORE-44 Junior Ho      29,99       2      0      2      59,98
            Stick
HAL8001     CCN Hockey Cap                 14,99      23     12     11     344,77
HAJ1002     CCN Z-55 Hockey Jersey         16,99       5      0      5      84,95
HPR4002     CCN T+2 Youth Hockey           19,99      13      2     11     259,87
            Elbow Pads
HPR5003     Reabuck 87K Junior Ice         44,99      13     12      1     584,87
            Hockey Pants
Page 2 of 2
                                                                           CU59144020
                                                                           2013-10-23
                                                                           2013-11-22

                                                                           INV2077448
                                                                           REP8396866
            Mcdermott
          es de Bellechasse School
            Court

          es de Bellechasse, QC
CA
29375-5022


Product     Description                  UnitPrice Ordered  B.O. Shipped   Total
HST1001     Bower RKT Junior Grip Hoc     129,99       2      0      2     259,98
            Stick
                                                 144     57     87    4496,47
                                                                        719,44
```

Extract Step

- Description
- Extraction Definition
  - Data Table: record.detail
  - ☑ Append values to current record
- Field Definition
  - Field List: Description
  - □ Add Unique ID to extraction field
  - Mode: Location
  - Left: 12
  - Right: 37
  - Top offset: 0
  - Height: 2
  - Post function:
  - Trim: Both
  - Type: String
  - Concatenation string:
  - □ Split
- Data format
  - No format

6. The extraction performed under the false branch of the second condition.



If the condition is false, the extraction is performed on single-line Description fields.

```
                                                                           INV2077448
                                                                           REP8396866
Mr Octavio Mcdermott
Saint-Charles de Bellechasse School
259, Renee Court

Saint-Charles de Bellechasse, QC
CA
29375-5022


Product     Description                  UnitPrice Ordered  B.O. Shipped   Total
HPR4003     Eastown SS32 Junior Hocke      15,98       9      1      8     143,82
            Elbow Pads
HPR3001     Bower Valor APX2 Junior H     104,99       2      1      1     209,98
            Hockey Gloves
HST2002     CCN Z-22 Junior Wood Hock      19,99      12      4      8     239,88
            Stick
HAH1001     Bower Hockey Cap               19,99      14      4     10     279,86
HST1002     Bower S1 Youth Hockey Sti      44,99      29      7     22    1304,71
HSK5001     Bower Valor APX3 Junior I     199,99       2      1      1     399,98
            Hockey skates
HAJ1003     CCN H22 Hockey Jersey          17,99      18     13      5     323,82
HST1005     Reabuck CORE-44 Junior Ho      29,99       2      0      2      59,98
            Stick
HAL8001     CCN Hockey Cap                 14,99      23     12     11     344,77
HAJ1002     CCN Z-55 Hockey Jersey         16,99       5      0      5      84,95
HPR4002     CCN T+2 Youth Hockey           19,99      13      2     11     259,87
            Elbow Pads
HPR5003     Reabuck 87K Junior Ice         44,99      13     12      1     584,87
            Hockey Pants
Page 2 of 2
                                                                           CU59144020
                                                                           2013-10-23
                                                                           2013-11-22

                                                                           INV2077448
                                                                           REP8396866

                                 e School

                                 e, QC
CA
29375-5022


Product     Description                  UnitPrice Ordered  B.O. Shipped   Total
HST1001     Bower RKT Junior Grip Hoc     129,99       2      0      2     259,98
            Stick
                                                 144     57     87    4496,47
                                                                        719,44
```

Extract Step

- Description
- Extraction Definition
  - Data Table: record.detail
  - ☑ Append values to current record
- Field Definition
  - Field List: Field1
  - □ Add Unique ID to extraction field
  - Mode: Location
  - Left: 12
  - Right: 37
  - Top offset: 0
  - Height: 1
  - Post function:
  - Trim: Both
  - Type: HTML String
  - Concatenation string: <br />
  - □ Split
- Data format
  - No format

For more information on how to add a step, please refer to Toolbar, Menus or Shortcut Keys under the **Interface** section.

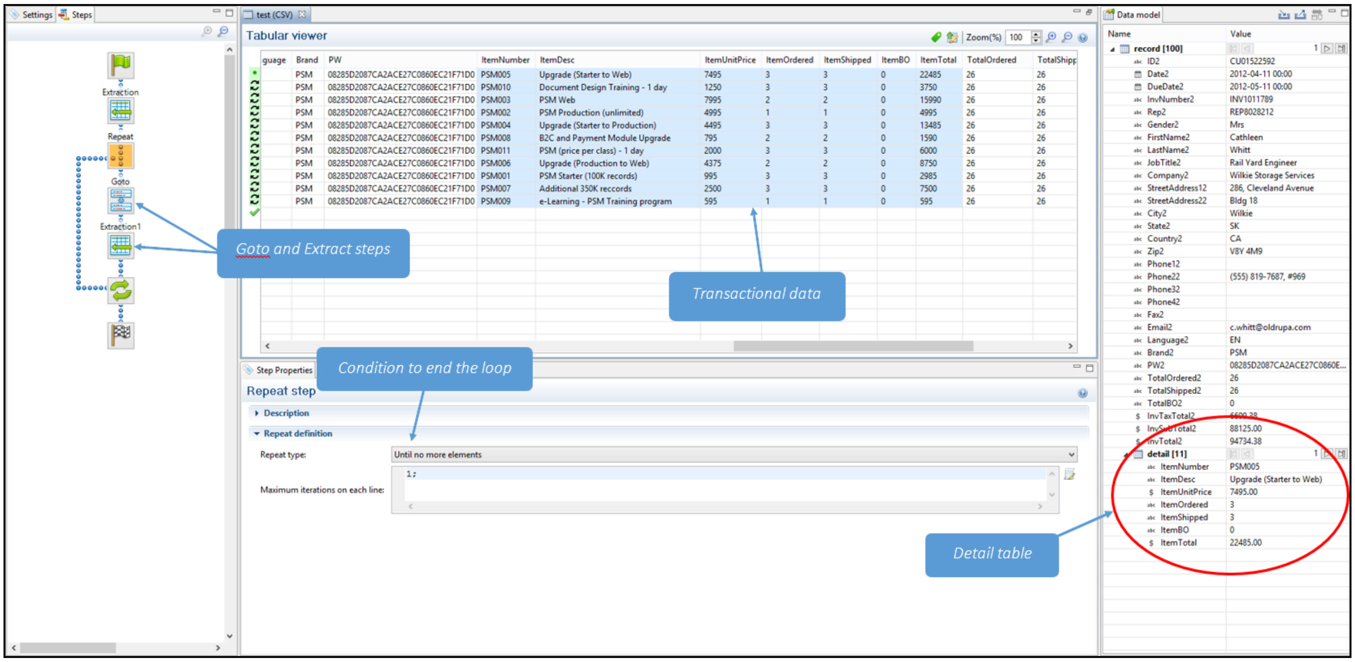
Properties

You can also further customize the step properties. Please refer to [The Step Properties Interface]() for more information.

Repeat

The **Repeat** step is a loop that may run 0 or more times, depending on the condition specified. It is generally used for the extraction of transactional data. A **Repeat** steps do not automatically move the pointer in the file. In order to avoid infinite loops, a **Goto** step must be present within the loop itself.

The following picture shows a usage example of a **Repeat** step. As you can see, the **Repeat** step is a loop that includes both **Goto** and **Extract** steps. The extraction result for transactional data is placed in a Detail table.



guage	Brand	PW	ItemNumber	ItemDesc	ItemUnitPrice	ItemOrdered	ItemShipped	ItemBO	ItemTotal	TotalOrdered	TotalShipp
	PSM	08285D2087CA2ACE27C0860EC21F71D0	PSM005	Upgrade (Starter to Web)	7495	3	3	0	22485	26	26
	PSM	08285D2087CA2ACE27C0860EC21F71D0	PSM010	Document Design Training - 1 day	1250	3	3	0	3750	26	26
	PSM	08285D2087CA2ACE27C0860EC21F71D0	PSM003	PSM Web	7995	2	2	0	15990	26	26
	PSM	08285D2087CA2ACE27C0860EC21F71D0	PSM002	PSM Production (unlimited)	4995	1	1	0	4995	26	26
	PSM	08285D2087CA2ACE27C0860EC21F71D0	PSM004	Upgrade (Starter to Production)	4495	3	3	0	13485	26	26
	PSM	08285D2087CA2ACE27C0860EC21F71D0	PSM008	B2C and Payment Module Upgrade	795	2	2	0	1590	26	26
	PSM	08285D2087CA2ACE27C0860EC21F71D0	PSM011	PSM (price per class) - 1 day	2000	3	3	0	6000	26	26
	PSM	08285D2087CA2ACE27C0860EC21F71D0	PSM006	Upgrade (Production to Web)	4375	2	2	0	8750	26	26
	PSM	08285D2087CA2ACE27C0860EC21F71D0	PSM001	PSM Starter (100K records)	995	3	3	0	2985	26	26
	PSM	08285D2087CA2ACE27C0860EC21F71D0	PSM007	Additional 350K reccords	2500	3	3	0	7500	26	26
	PSM	08285D2087CA2ACE27C0860EC21F71D0	PSM009	e-Learning - PSM Training program	595	1	1	0	595	26	26

Step Properties

Repeat step

▸ Description

▾ Repeat definition

Repeat type: Until no more elements

Maximum iterations on each line: 1;

Name	Value
record [100]	1
ID2	CU01522592
Date2	2012-04-11 00:00
DueDate2	2012-05-11 00:00
InvNumber2	INV1011789
Rep2	REP8028212
Gender2	Mrs
FirstName2	Cathleen
LastName2	Whitt
JobTitle2	Rail Yard Engineer
Company2	Wilkie Storage Services
StreetAddress12	286, Cleveland Avenue
StreetAddress22	Bldg 18
City2	Wilkie
State2	SK
Country2	CA
Zip2	V8Y 4M9
Phone12	
Phone22	(555) 819-7687, #969
Phone32	
Phone42	
Fax2	
Email2	c.whitt@oldrupa.com
Language2	EN
Brand2	PSM
PW2	08285D2087CA2ACE27C0860E...
TotalOrdered2	26
TotalShipped2	26
TotalBO2	0
InvTaxTotal2	6609.38
InvSubTotal2	88125.00
InvTotal2	94734.38
detail [11]	1
ItemNumber	PSM005
ItemDesc	Upgrade (Starter to Web)
ItemUnitPrice	7495.00
ItemOrdered	3
ItemShipped	3
ItemBO	0
ItemTotal	22485.00

For more information on how to add a step, please refer to [Toolbar](), [Menus]() or [Shortcut Keys]() under the **Interface** section.

Properties

You can also further customize the step properties. Please refer to [The Step Properties Interface]() for more information.

Note

By default, if an **Extract** step is added within a **Repeat** step, its extraction is made in a detail table.

Note

If an XML node that has children is selected, and the pointer is currently at this node, creating a repeat step will loop on that node.

Extract Field

The **Add Extract Field** function adds the selected data to a selected **Extract** step in the **Steps** pane. If multiple lines, nodes or fields are selected, multiple extract fields are added simultaneously (see also About Records and Fields for more information).

For more information on how to add a step, please refer to Toolbar, Menus or Shortcut Keys under the **Interface** section.

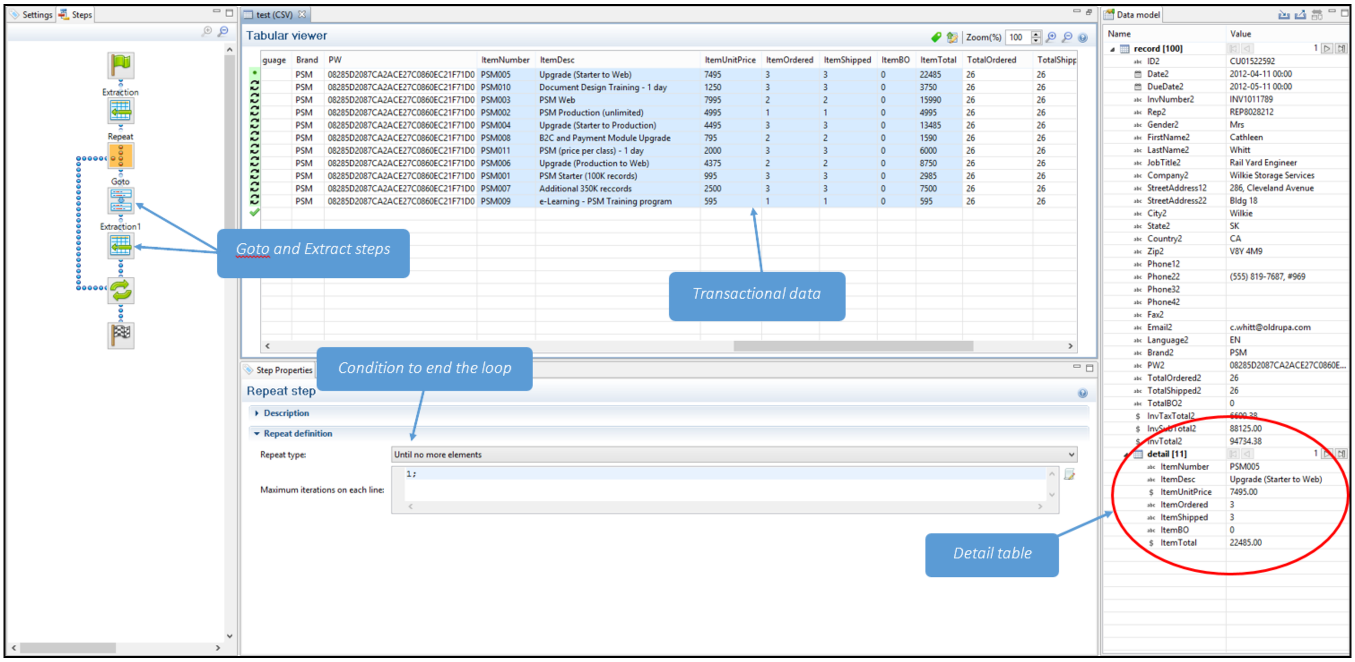
Properties

You can also further customize the step properties. Please refer to The Step Properties Interface for more information.

Action

The **Action** step can set the value for a Source Record property or execute a JavaScript code. The **Action** Step can run multiple specific actions one after the other in order.

The following picture shows a usage example of an **Action** step...



guage	Brand	PW	ItemNumber	ItemDesc	ItemUnitPrice	ItemOrdered	ItemShipped	ItemBO	ItemTotal	TotalOrdered	TotalShipp
	PSM	08285D2087CA2ACE27C0860EC21F71D0	PSM005	Upgrade (Starter to Web)	7495	3	3	0	22485	26	26
	PSM	08285D2087CA2ACE27C0860EC21F71D0	PSM010	Document Design Training - 1 day	1250	3	3	0	3750	26	26
	PSM	08285D2087CA2ACE27C0860EC21F71D0	PSM003	PSM Web	7995	2	2	0	15990	26	26
	PSM	08285D2087CA2ACE27C0860EC21F71D0	PSM002	PSM Production (unlimited)	4995	1	1	0	4995	26	26
	PSM	08285D2087CA2ACE27C0860EC21F71D0	PSM004	Upgrade (Starter to Production)	4495	3	3	0	13485	26	26
	PSM	08285D2087CA2ACE27C0860EC21F71D0	PSM008	B2C and Payment Module Upgrade	795	2	2	0	1590	26	26
	PSM	08285D2087CA2ACE27C0860EC21F71D0	PSM011	PSM (price per class) - 1 day	2000	3	3	0	6000	26	26
	PSM	08285D2087CA2ACE27C0860EC21F71D0	PSM006	Upgrade (Production to Web)	4375	2	2	0	8750	26	26
	PSM	08285D2087CA2ACE27C0860EC21F71D0	PSM001	PSM Starter (100K records)	995	3	3	0	2985	26	26
	PSM	08285D2087CA2ACE27C0860EC21F71D0	PSM007	Additional 350K reccords	2500	3	3	0	7500	26	26
	PSM	08285D2087CA2ACE27C0860EC21F71D0	PSM009	e-Learning - PSM Training program	595	1	1	0	595	26	26

Repeat step

Description

Repeat definition

Repeat type: Until no more elements

Maximum iterations on each line: 1;

Name	Value
record [100]	1
ID2	CU01522592
Date2	2012-04-11 00:00
DueDate2	2012-05-11 00:00
InvNumber2	INV1011789
Rep2	REP8028212
Gender2	Mrs
FirstName2	Cathleen
LastName2	Whitt
JobTitle2	Rail Yard Engineer
Company2	Wilkie Storage Services
StreetAddress12	286, Cleveland Avenue
StreetAddress22	Bldg 18
City2	Wilkie
State2	SK
Country2	CA
Zip2	V8Y 4M9
Phone12	
Phone22	(555) 819-7687, #969
Phone32	
Phone42	
Fax2	
Email2	c.whitt@oldrupa.com
Language2	EN
Brand2	PSM
PW2	08285D2087CA2ACE27C0860E...
TotalOrdered2	26
TotalShipped2	26
TotalBO2	0
InvTaxTotal2	6609.38
InvSubTotal2	88125.00
InvTotal2	94734.38
detail [11]	1
ItemNumber	PSM005
ItemDesc	Upgrade (Starter to Web)
ItemUnitPrice	7495.00
ItemOrdered	3
ItemShipped	3
ItemBO	0
ItemTotal	22485.00

For more information on how to add a step, please refer to Toolbar, Menus or Shortcut Keys under the **Interface** section.

Properties

You can also further customize the step properties. Please refer to The Step Properties Interface for more information.

Data Model

A Data Model is the format into which Records are saved. It is also a file format that can be exchanged between different parts of the PReS Connect solution.

Data Model Principle

The idea behind the Data Model is that the structure of a Record can be separated from the actual data. It can be shared between multiple templates, data mapping configurations, job presets and output presets. These files are deemed **compatible** if their data model is identical or if one data model is an exact subset of another.

Data Models can be used directly in the Designer module, even if no data is attached to it, to create the template.

They can be imported to, and exported from, the Data Model Pane in the Designer and DataMapper Modules.

For more information about the operations you can perform on the Data Model, please refer to The Data Model Interface.

About Records and Fields

A Record is a collection of fields and contains all the data about one particular person, company, or item in a database that is required for creating a single document for a single recipient (see Record for more information).

A **Field** is part of a record and contains a single piece of data for the subject of the record. If the record is an employee, a field would be Firstname, Lastname, or City or State.

Data Model File Structure

The Data Model file is a XML file that contains the structure of the data model, including each field's name, data type, and any number of detail tables and nested tables.

Example for promotional data, including a simple name and address:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<datamodel schemaVersion="1.0.0.3" name="Generic Address Block"
version="1"
xmlns="http://www.objectiflune.com/connectschemas/DataModelConfig"
xsi:schemaLocation="http://www.objectiflune.com/connectschemas/Data
ModelConfig
http://www.objectiflune.com/connectschemas/DataModelConfig/1_0_0_
3.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <field type="string" name="Name" required="true"/>
    <field type="string" name="Organization" required="true"/>
    <field type="string" name="Address1" required="true"/>
    <field type="string" name="Address2" required="true"/>
    <field type="string" name="Address3" required="true"/>
    <field type="string" name="City" required="true"/>
    <field type="string" name="StateOrProvince" required="true"/>
    <field type="string" name="Country" required="true"/>
    <field type="string" name="ZipOrPostalCode" required="true"/>
    <field type="string" name="Extra1" required="true"/>
    <field type="string" name="Extra2" required="true"/>
</datamodel>
```

Example with transactional details, a simple invoice format

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<datamodel schemaVersion="1.0.0.3" name="Transactional Invoice"
version="1"
xmlns="http://www.objectiflune.com/connectschemas/DataModelConfig"
xsi:schemaLocation="http://www.objectiflune.com/connectschemas/Data
ModelConfig
http://www.objectiflune.com/connectschemas/DataModelConfig/1_0_0_
3.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <field type="string" name="ID" required="true"/>
    <field type="string" name="Gender" required="true"/>
    <field type="string" name="LastName" required="true"/>
    <field type="string" name="FirstName" required="true"/>
    <field type="string" name="Address1" required="true"/>
    <field type="string" name="Address2" required="true"/>
    <field type="string" name="City" required="true"/>
    <field type="string" name="State" required="true"/>
    <field type="string" name="Country" required="true"/>
    <field type="string" name="ZipCode" required="true"/>
    <field type="string" name="Title" required="true"/>
    <field type="string" name="Company" required="true"/>
    <field type="string" name="Phone2" required="true"/>
    <field type="string" name="Email" required="true"/>
    <field type="string" name="Language" required="true"/>
    <field type="string" name="Number" required="true"/>
    <field type="datetime" name="Date" required="true"/>
    <field type="datetime" name="DueDate" required="true"/>
    <field type="string" name="Rep" required="true"/>
    <field type="currency" name="SubTotal" required="true"/>
    <field type="currency" name="TaxTotal" required="true"/>
    <field type="currency" name="Total" required="true"/>
    <table name="detail">
        <field type="string" name="Number2" required="true"/>
        <field type="string" name="Description" required="true"/>
        <field type="currency" name="UnitPrice" required="true"/>
        <field type="integer" name="Ordered" required="true"/>
        <field type="integer" name="Shipped" required="true"/>
        <field type="integer" name="BackOrder" required="true"/>
        <field type="currency" name="Total2" required="true"/>
    </table>
</datamodel>
```

Example of Nested Tables (one table into another)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```
<datamodel schemaVersion="1.0.0.3" name="Nested Table Example"
version="1"
xmlns="http://www.objectiflune.com/connectschemas/DataModelConfig"
xsi:schemaLocation="http://www.objectiflune.com/connectschemas/Data
ModelConfig
http://www.objectiflune.com/connectschemas/DataModelConfig/1_0_0_
3.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <field type="string" name="RecordField" required="true"/>
    <table name="details">
        <field type="string" name="DetailsTableField"
required="true"/>
        <table name="nestedtable">
            <field type="string" name="NestedTableField"
required="true"/>
        </table>
    </table>
</datamodel>
```

Example of Default Values that can be added to any field with the defaultValue attribute:

```
<field type="string" name="RecordField" required="true"
 defaultValue="My Default Value">
```

Data Source (Settings)

A data source is simply the source of the data. It can be a file (CSV, PDF, TXT, XML) or a particular database. The data might be located on the same computer as the program, or on another computer somewhere on a network.

The first step to do before creating a Data Mapping configuration is to set up **Input Data** (setting the delimiters) and **Boundaries** (setting the trigger) in the **Settings** pane.

Input Data (Delimiters)

Delimiters defines the separation within the source file. For example, in a CSV file the delimiters are between each row whereas in a PDF file, delimiters separate pages.

Please refer to The Settings Pane Interface for more information about each fields and buttons.

Boundaries

Boundaries are the division between source records. They can be set using different types of rules such as text, pages or even custom JavaScript rules.

Boundaries differ from Delimiters. When defining boundaries , multiple delimiters can be included between boundaries - many pages per invoice, many CSV row per transaction, etc.

To set a boundary, a specific *trigger* must be found. The trigger is something that is either static ("Page 1 of" on text or PDF files) or something that changes on each source record (a customer ID at a specific location, a username, etc).

Please refer to The Settings Pane Interface for more information about each fields and buttons.

Data Samples

The Data Samples are the data sources that have been imported into the data mapping configuration.

Please refer to The Settings Pane Interface for more information about each fields and buttons.

External JS Libraries

You can use the External JS libraries to add more JavaScript functionality to your data mapping configuration. Any functions included in the JS library will be available in Preprocessor scripts, as well as Action tasks, Post Functions and JavaScript -based extraction steps.

For example let's take the following JavaScript file, for example:

```
function myAddFunction(p1, p2) {
    return p1 + p2;
};
```

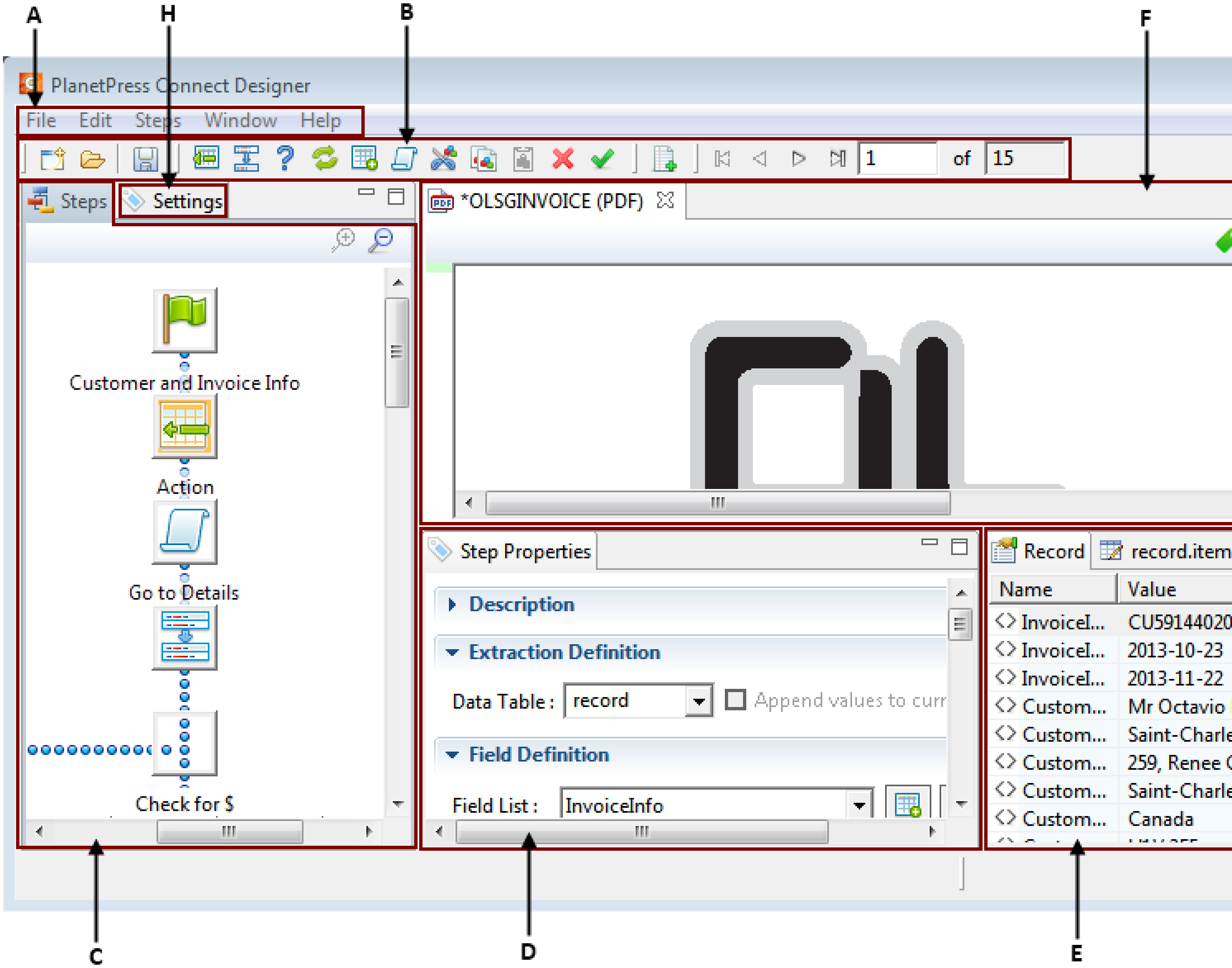
If this is saved as myFunction.js and imported, then the following would work anywhere in the configuration:

```
var result = myAddFunction(25, 12); // returns 37!
```

The External JS Libraries box displays all the libraries that have been imported into the data mapping configuration.

Please refer to The Settings Pane Interface for more information about each fields and buttons.

DataMapper User Interface



Data Table : record

□ Append values to curr

Field List : InvoiceInfo

Name	Value
InvoiceI...	CU59144020
InvoiceI...	2013-10-23
InvoiceI...	2013-11-22
Custom...	Mr Octavio
Custom...	Saint-Charle
Custom...	259, Renee C
Custom...	Saint-Charle
Custom...	Canada

Interface Sections and Panels

A. Menus
B. Toolbars
C. Steps Pane
D. Step Properties Pane

E. Data Model Pane
F. Data Viewer
G. Messages Pane
H. Settings Pane

Menus

The following menu items are shown in the DataMapper Module's menu:

File Menu

- **New...** : Opens the Creating a New Data Mapping Configuration dialog.
- **Open**: Opens a standard File Open dialog. This dialog can be used to open Templates and data mapping configurations.
- **Open Recent**: List the most recently opened Templates and configurations. Clicking on a template will open it in the Designer module, clicking on a data mapping configuration will open it in the DataMapper module.
- **Close**: Close the currently open data mapping configuration or Template. If the file needs to be saved, the appropriate Save dialog will open.
- **Close All**: Close any open data mapping configuration or Template. If any of the files need to be saved, the Save Resources dialog opens.
- **Save**: Saves the current data mapping configuration or Template to its current location on disk. If the file is a data mapping configuration and has never been saved, the Save As dialog appears instead.
- **Save As...**: Saves the current data mapping configuration or Template to a new location on disk. In the case of Templates, it is saved to a location that can be different than the local repository.
- **Save All**: Saves all open files. If any of the open files have never been saved, the Save As dialog opens for each new unsaved file.
- **Revert**: Appears only in the Designer module. Reverts all changes to the state in which the file was opened or created.
- **Add Data**: Adds data either to the current data mapping configuration or to the open template. In data mapping configuration
  - **From File...**: Opens the dialog to add a new data file to the currently loaded data mapping configuration. Not available if the currently loaded data mapping configuration connects to a database source.
  - **From Database...**: Opens the Edit Database Configuration dialog. Not available if the currently loaded data mapping configuration is file-based.

- **Send to Workflow**: Opens theSend to Workflow dialog to send files to a local PReS Workflow software installation.
- **Exit:**Closes the software. If any of the files need to be saved, the Save Resources dialog opens.

Edit Menu

- **Undo**: Undoes the previous action.
- **Redo**: Redoes the last action that was undone.
- **Cut Step**: Removes the currently selected step and places it in the clipboard. If the step is a Repeat or a Condition, all steps under it are also placed in the clipboard. If there is already a step in the clipboard, it will be overwritten.
- **Copy Step**: Places a copy of the currently selected step in the clipboard. The same details as the Cut step applies.
- **Paste Step**: Takes the step or steps in the clipboard and places them in the Steps after the currently selected step.
- **Delete Step**: Deletes the currently selected step. If the step is a Repeat or Condition, all steps under it are also deleted.
- **Cut**: Click to remove the currently selected step, or steps, and place them in the clipboard.
- **Copy**: Click to place a copy of the currently selected step, or steps, in the clipboard.
- **Paste**: Click to place any step, or steps, from the clipboard before the currently selected step in the Steps Pane.

Data Menu

- **Hide/Show datamap**: Click to show or hide the icons to the left of the Data Viewer that displays how the steps affect the line.
- **Hide/Show extracted data**: Click to show or hide the extraction selections indicating that data is extracted. This simplifies making data selections in the same areas and is useful to display the original data.
- **Validate All Records**: Runs the Steps on all records and verifies that no errors are present in any of the records. Errors are displayed in the Messages Pane.

Steps

- **Ignore Step**: Click to set the step to be ignored (aka disabled). Disabled steps do not run when in DataMapper and do not execute when the data mapping configuration is executed in Workflow. However, they can still be modified normally.
- **Add Extract Step**: Adds an Extract Step with one or more extract fields. If more than one line or field is selected in the Data Viewer, each line or field will have an extract field.

- **Add Goto Step**: Adds a Goto step that moves the selection pointer to the beginning of the data selection. For instance if an XML node is selected, the pointer moves to where that node is located.
- **Add Condition Step**: Adds a condition based on the current data selection. The "True" branch gets run when the text is found on the page. Other conditions are available in the step properties once it has been added.
- **Add Repeat Step**: Adds a loop that is based on the current data selection, and depending on the type of data. XML data will loop on the currently selected node, CSV loops for all rows in the record. In Text and PDF data, if the data selection is on the same line as the cursor position, the loop will be for each line until the end of the record. If the data selection is on a lower line, the loop will be for each line until the text in the data selection is found at the specified position on the line (e.g. until "TOTAL" is found).
- **Add Extract Field**: Adds the data selection to the selected Extract step, if an extract step is currently selected. If multiple lines, nodes or fields are selected, multiple extract fields are added simultaneously.
- **Add Action Step**: Adds a step to run one or more specific actions such as running a JavaScript expression or setting the value of a Source Record Property.

View Menu

- **Zoom In**: Click to zoom in the Steps Pane.
- **Zoom Out**: Click to zoom out the Steps Pane.

Window Menu

- **Show View**
  - **Messages**: Shows the Messages Pane.
  - **Steps**: Shows the Steps Pane.
  - **Settings**: Shows the Settings Pane.
  - **Record**: Shows the Record Pane.
  - **Detail tables** : Each detail table and nested table is listed here. Click on one to show it in the Data Model Pane.
  - **Step Properties**: Shows the Step Properties Pane.
- **Reset Perspective**: Resets all toolbars and panes to the initial configuration of the module.

Help Menu

- **Software Activation**: Displays the Software Activation dialog. See Activating your license.
- **Help Topics**: Click to open this documentation.
- **Contact Support**: Click to open the Objectif Lune Contact Page in the default system Web browser.
- **About PReS Connect Designer**: Displays the software's About dialog.
- **Welcome Screen**: Click to re-open the Welcome Screen.

Panes

The following topics explain in detail the different panes of the DataMapper Interface.

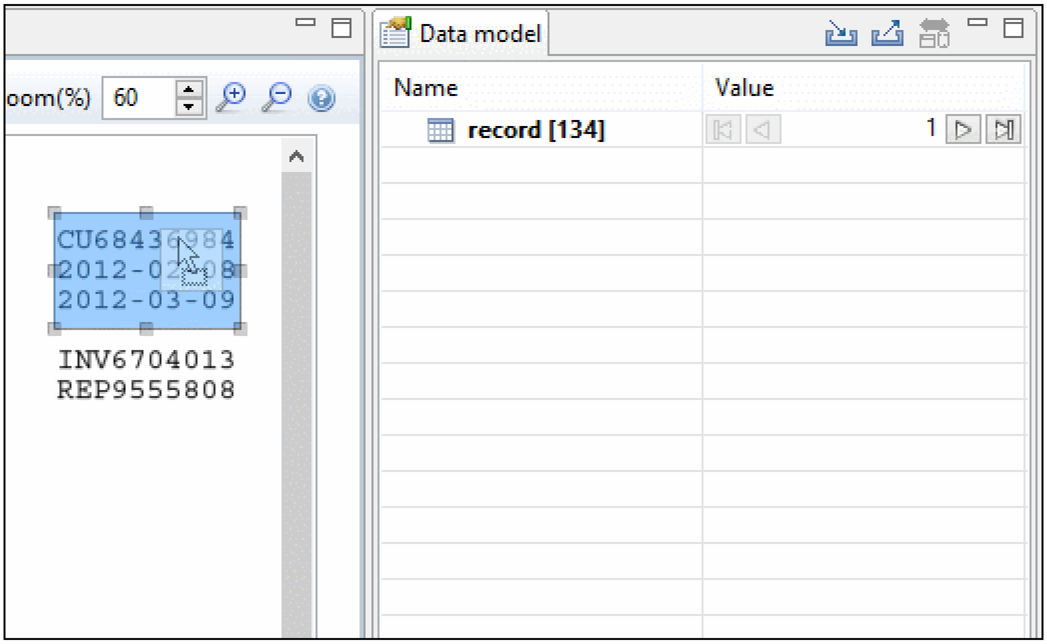
Data Model Pane

The Data Model Pane displays the Extracted Record, all its fields as well as any detail tables present in the Extracted Record. The information shown is the extracted information for the current record within the Record Set. It is also used as a navigation tool between records and all tables.

Data is displayed as a tree view, with the root level being the Record table, levels below it being detail tables, and any level below being called Nested Tables.

Adding Data

Beyond the methods for adding data described in the Steps Pane and the **Data Viewer**, there is a specific way to add an **Extract** step directly into the Data Model pane, which is drag & drop: Dragging a data selection (text and PDF) or a field (CSV, XML and Database) into the Data Model pane will automatically add the appropriate extraction method into a specific location.



There are some specific tricks and limitations, however:

- If no **Extract** step is currently selected in the Steps Pane, a new **Extract** step is created. If an **Extract** step is selected, new **Extract Fields** are added to this existing step.
- Dragging a data selection or fields into a specific level of the Data Model (Record or detail tables) will add the fields to that level.
- When dragging data into a detail table, the Extract Step must be located within the appropriate Repeat Step, otherwise the extract will not function properly.
- If a Data Model is loaded into the Data Model Pane, when dragging named fields (CSV, XML, Database), if a field of the same name exists in that level, the extract field extract the data to that field, otherwise it creates a new one.
- Dragging a single field onto a Data Model field will extract to that field regardless of name.

Modifying the Data Model

It is possible to modify the fields within the Data Model pane, effectively generating data models, even if no data is present in the pane. To do this, use the contextual menu within the pane itself. The following options appear depending on the selected line when right-clicking:

- **Add a field**: Click to add a new field at the current level (record or table). Enter the field name in the dialog and click OK to add it.
- **Add a table**: Click to add a new table at the current level (record or existing table). Enter the table name in the dialog and click OK to add it.

- **Rename**: Click to rename the selected table or field. Enter the new name and click OK to rename. Only available for empty Data Model Fields (see below) tables that only contain them.
- **Delete**: Click to delete the selected table or field. Only available for empty Data Model Fields or tables that only contain them.
- **Set Type**: Use the list to select which the field type (see Data Types). Only available for empty Data Model Fields or tables that only contain them.
- **Default Value**: Click to set the default value for empty Data Model Fields. This value is used if no extraction is present, or if an extraction attached to this field returns no value.

Field Display

Fields in the Data Model pane are displayed in specific ways to simplify comprehension of the display data:

- **Value**: The current value of the extracted field, based on the current Source Record.
- The column on the left displays the **name** of the field. The column on the right displays the **extracted data** if it exists.
- The column on the right displays the current **value**of the extracted field based on the current Source Record, when anExtract Step has an extraction for this field.
- The **icon** to the left of the name indicates the **Data Type** of the field (see Data Types).
- A field name with an **asterisk** to the right indicates that this field is part of an imported Data Model file (called a **Data Model Field**).
- A field with a **grey background** indicates this Data Model Field does not have any attached extracted data.
- A field with a **white background** indicates that the field has attached extracted data but the step extracting the data is not currently selected.
- A field with a **blue background** indicates that the field has attached extracted data and the step extracting the data is currently selected.

Record Navigation

Records can be navigated directly from the Data Model Pane. The default Record level navigates between records both in the Data Model pane and the Data Viewer, while each table has a similar navigation that influence that table and each under it.

- **Expand/Contract**: Click to hide or show any fields or tables under the current table level.
- **Table Name**: Displays the name of the table as well as the number of records at that level (in [brackets]). At the Record level this is the number of Source records (affected by the

Boundary settings, Preprocessor and the Record Limit setting). In other levels it represents the number of entries in a table.

- **Number of Records**: The number of available records in the Data Sample. Affected by the Boundary settings, Preprocessor and the Record Limit setting.
- **First Record**: Jumps to the first record in the Data Sample. Disabled if the first record is already shown.
- **Previous Record**: Moves to the previous record in the Data Sample. Disabled if the first record is already shown.
- **Current Record**: Displays the current record or table entry. Type in a record number and press the **Enter** key to display that record. The number has to be within the number of available records in the Data Sample.
- **Next Record**: Moves to the next record in the Data Sample. Disabled if the last record is already shown.
- **Last Record**: Jumps to the last record in the Data Sample. Disabled if the last record is already shown. If a Record Limit is set in the Settings Pane, the last record will be within that limit.

Importing/Exporting Data Models

Within the Data Model pane, data model files can be exported using the current structure, or imported in order to simplify the extraction process.

Importing and Exporting Data Models is done from within the Data model Pane, using the top-right icons 📥 and 📤.

Rules for importing:

- Imported Data Model fields always overwrite existing field properties when the field name is the same. Non-existent fields are created automatically with the appropriate field settings. The import is case-insensitive, so fields with a different case will be ignored.
- All data model fields are tagged with the Asterisk (*). If overwriting an existing data model, existing data model fields that are not in the imported data model file will have their asterisks removed.

Detail Tables

Detail tables contain transactional data that is created when an **Extract** step is added within a **Repeat** step.

In the most basic of transactional systems, a single detail table is used. However, it is possible to create multiple detail tables, as well as nested tables. When detail and nested tables are present in the record, they are displayed as separate levels in the Data Model (see The Data Model Interface for more information).

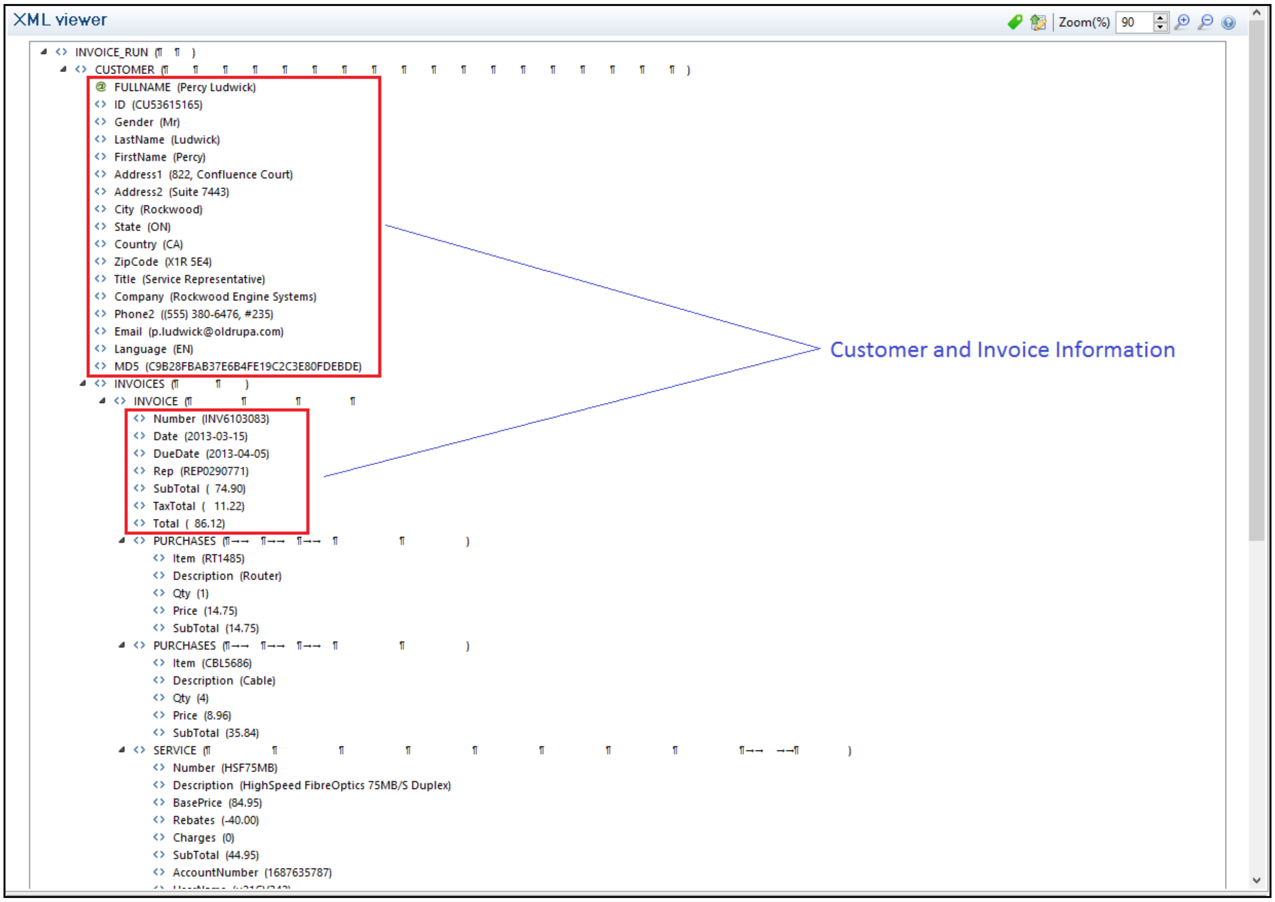
Multiple detail tables

Multiple detail tables are useful when more than one type of transactional data is necessary, for instance if two different type of fields are used for different types of data. One example of this would be invoice data containing purchases (items with a set price, quantity, item number) as well as services (service with a price, frequency, contract end date, etc).

Creating more than one detail table is simple. Click the **Extract** steps and change the name of tables from **record.details** to something else. In the following example, we will create two tables called **record.purchases** and **record.services**.

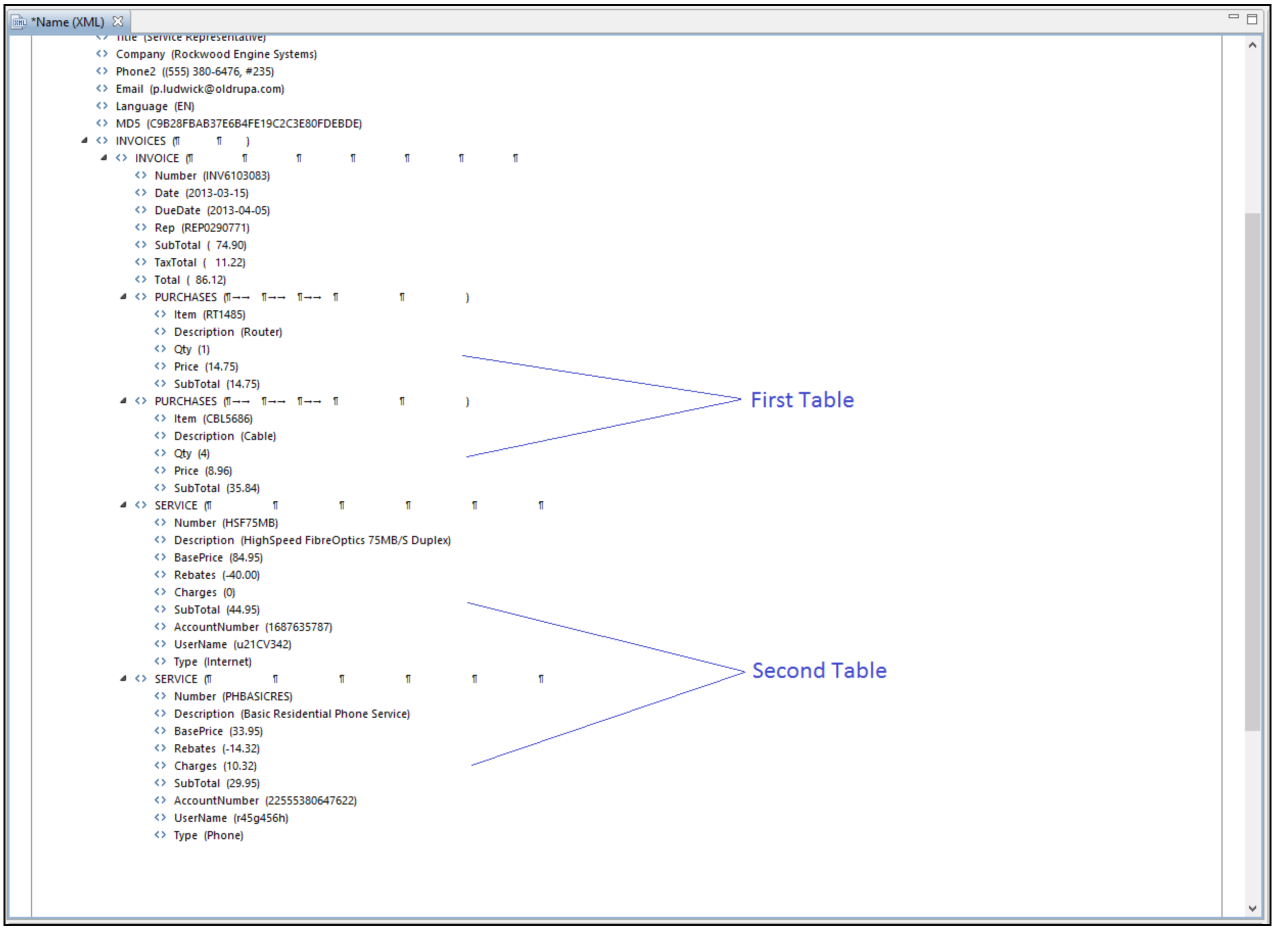
Example

1. Data Overview.



```
◢ <> INVOICE_RUN (¶  ¶  )
  ◢ <> CUSTOMER (¶  ¶  ¶  ¶  ¶  ¶  ¶  ¶  ¶  ¶  ¶  ¶  ¶  ¶  ¶  ¶  ¶  ¶  )
      @ FULLNAME  (Percy Ludwick)
      <> ID  (CU53615165)
      <> Gender  (Mr)
      <> LastName  (Ludwick)
      <> FirstName  (Percy)
      <> Address1  (822, Confluence Court)
      <> Address2  (Suite 7443)
      <> City  (Rockwood)
      <> State  (ON)
      <> Country  (CA)
      <> ZipCode  (X1R 5E4)
      <> Title  (Service Representative)
      <> Company  (Rockwood Engine Systems)
      <> Phone2  ((555) 380-6476, #235)
      <> Email  (p.ludwick@oldrupa.com)
      <> Language  (EN)
      <> MD5  (C9B28FBAB37E6B4FE19C2C3E80FDEBDE)
    ◢ <> INVOICES (¶        ¶     )
      ◢ <> INVOICE (¶           ¶          ¶          ¶
          <> Number  (INV6103083)
          <> Date  (2013-03-15)
          <> DueDate  (2013-04-05)
          <> Rep  (REP0290771)
          <> SubTotal  (  74.90)
          <> TaxTotal  (   11.22)
          <> Total  (  86.12)
        ◢ <> PURCHASES (¶→→  ¶→→  ¶→→  ¶           ¶           )
            <> Item  (RT1485)
            <> Description  (Router)
            <> Qty  (1)
            <> Price  (14.75)
            <> SubTotal  (14.75)
        ◢ <> PURCHASES (¶→→  ¶→→  ¶→→  ¶           ¶           )
            <> Item  (CBL5686)
            <> Description  (Cable)
            <> Qty  (4)
            <> Price  (8.96)
            <> SubTotal  (35.84)
        ◢ <> SERVICE (¶           ¶          ¶          ¶          ¶          ¶          ¶          ¶→→  →→¶          )
            <> Number  (HSF75MB)
            <> Description  (HighSpeed FibreOptics 75MB/S Duplex)
            <> BasePrice  (84.95)
            <> Rebates  (-40.00)
            <> Charges  (0)
            <> SubTotal  (44.95)
            <> AccountNumber  (1687635787)
```

Customer and Invoice Information

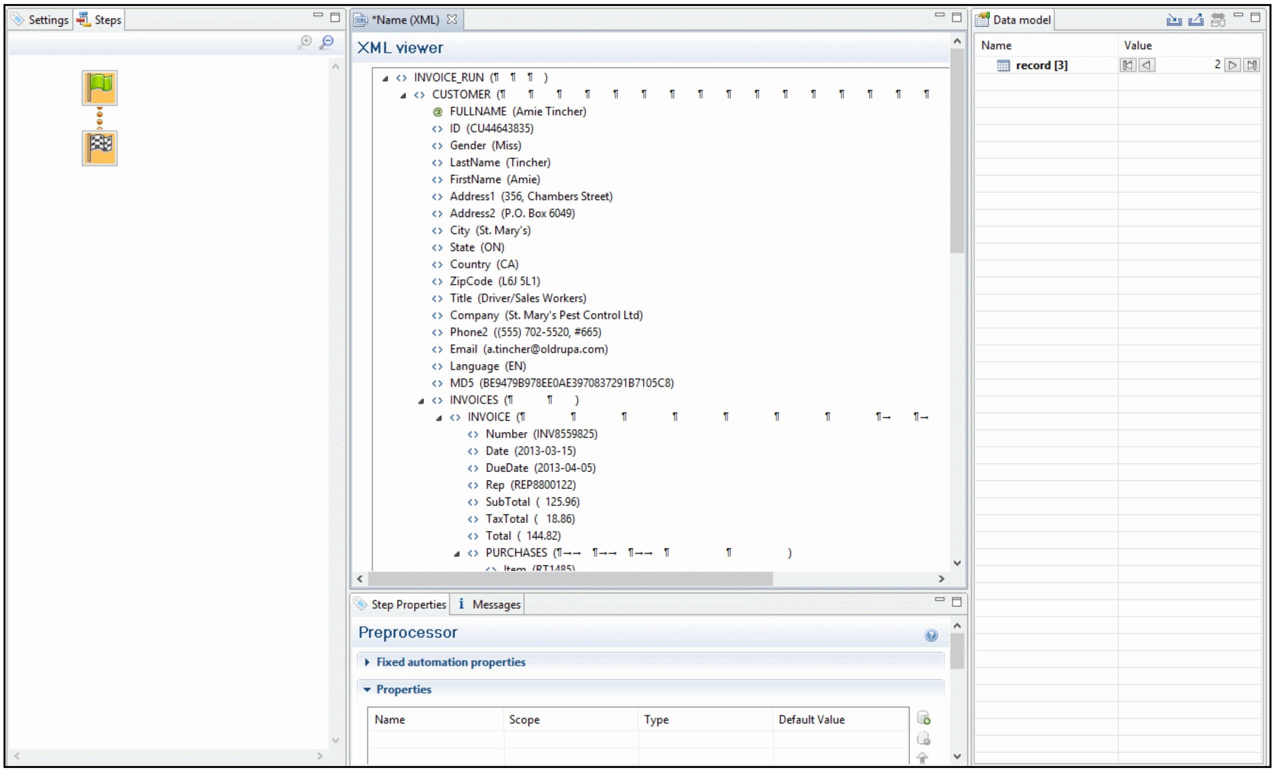


```
      <> Title  (Service Representative)
      <> Company  (Rockwood Engine Systems)
      <> Phone2  ((555) 380-6476, #235)
      <> Email  (p.ludwick@oldrupa.com)
      <> Language  (EN)
      <> MD5  (C9B28FBAB37E6B4FE19C2C3E80FDEBDE)
    ◢ <> INVOICES  (¶        ¶      )
      ◢ <> INVOICE  (¶         ¶         ¶         ¶         ¶         ¶         ¶
          <> Number  (INV6103083)
          <> Date  (2013-03-15)
          <> DueDate  (2013-04-05)
          <> Rep  (REP0290771)
          <> SubTotal  (  74.90)
          <> TaxTotal  (   11.22)
          <> Total  (  86.12)
        ◢ <> PURCHASES  (¶→→   ¶→→   ¶→→   ¶           ¶           )
            <> Item  (RT1485)
            <> Description  (Router)
            <> Qty  (1)
            <> Price  (14.75)
            <> SubTotal  (14.75)
        ◢ <> PURCHASES  (¶→→   ¶→→   ¶→→   ¶           ¶           )
            <> Item  (CBL5686)
            <> Description  (Cable)
            <> Qty  (4)
            <> Price  (8.96)
            <> SubTotal  (35.84)
        ◢ <> SERVICE  (¶           ¶           ¶           ¶           ¶           ¶
            <> Number  (HSF75MB)
            <> Description  (HighSpeed FibreOptics 75MB/S Duplex)
            <> BasePrice  (84.95)
            <> Rebates  (-40.00)
            <> Charges  (0)
            <> SubTotal  (44.95)
            <> AccountNumber  (1687635787)
            <> UserName  (u21CV342)
            <> Type  (Internet)
        ◢ <> SERVICE  (¶           ¶           ¶           ¶           ¶           ¶
            <> Number  (PHBASICRES)
            <> Description  (Basic Residential Phone Service)
            <> BasePrice  (33.95)
            <> Rebates  (-14.32)
            <> Charges  (10.32)
            <> SubTotal  (29.95)
            <> AccountNumber  (22555380647622)
            <> UserName  (r45g456h)
            <> Type  (Phone)
```

First Table

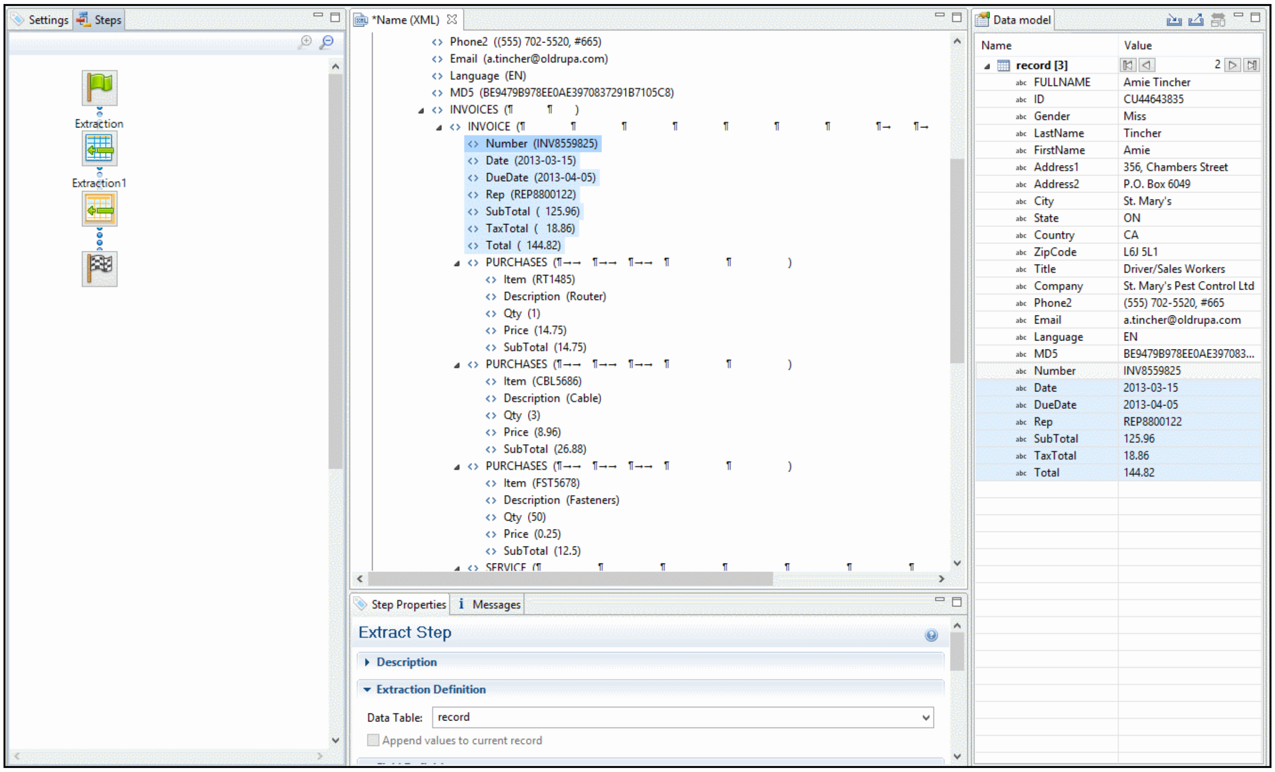
Second Table

2. Extracting Customer and Invoice information.



```
INVOICE_RUN (¶ ¶ ¶ )
  CUSTOMER (¶ ¶ ¶ ¶ ¶ ¶ ¶ ¶ ¶ ¶ ¶ ¶ ¶ ¶ ¶ ¶
    @ FULLNAME (Amie Tincher)
    ID (CU44643835)
    Gender (Miss)
    LastName (Tincher)
    FirstName (Amie)
    Address1 (356, Chambers Street)
    Address2 (P.O. Box 6049)
    City (St. Mary's)
    State (ON)
    Country (CA)
    ZipCode (L6J 5L1)
    Title (Driver/Sales Workers)
    Company (St. Mary's Pest Control Ltd)
    Phone2 ((555) 702-5520, #665)
    Email (a.tincher@oldrupa.com)
    Language (EN)
    MD5 (BE9479B978EE0AE3970837291B7105C8)
    INVOICES (¶ ¶ )
      INVOICE (¶ ¶ ¶ ¶ ¶ ¶ ¶ ¶→ ¶→
        Number (INV8559825)
        Date (2013-03-15)
        DueDate (2013-04-05)
        Rep (REP8800122)
        SubTotal ( 125.96)
        TaxTotal ( 18.86)
        Total ( 144.82)
        PURCHASES (¶→→ ¶→→ ¶→→ ¶ ¶ )
          Item (RT1485)
```

3. Extracting Transactional Data.



```
<> Phone2  ((555) 702-5520, #665)
<> Email  (a.tincher@oldrupa.com)
<> Language  (EN)
<> MD5  (BE9479B978EE0AE3970837291B7105C8)
<> INVOICES  (¶        ¶      )
  <> INVOICE  (¶       ¶       ¶       ¶       ¶       ¶       ¶       ¶→     ¶→
    <> Number  (INV8559825)
    <> Date  (2013-03-15)
    <> DueDate  (2013-04-05)
    <> Rep  (REP8800122)
    <> SubTotal  ( 125.96)
    <> TaxTotal  (  18.86)
    <> Total  ( 144.82)
    <> PURCHASES  (¶→→  ¶→→  ¶→→  ¶          ¶          )
      <> Item  (RT1485)
      <> Description  (Router)
      <> Qty  (1)
      <> Price  (14.75)
      <> SubTotal  (14.75)
    <> PURCHASES  (¶→→  ¶→→  ¶→→  ¶          ¶          )
      <> Item  (CBL5686)
      <> Description  (Cable)
      <> Qty  (3)
      <> Price  (8.96)
      <> SubTotal  (26.88)
    <> PURCHASES  (¶→→  ¶→→  ¶→→  ¶          ¶          )
      <> Item  (FST5678)
      <> Description  (Fasteners)
      <> Qty  (50)
      <> Price  (0.25)
      <> SubTotal  (12.5)
    <> SERVICE  (¶        ¶        ¶        ¶        ¶        ¶        ¶
```

Name	Value
record [3]	2
FULLNAME	Amie Tincher
ID	CU44643835
Gender	Miss
LastName	Tincher
FirstName	Amie
Address1	356, Chambers Street
Address2	P.O. Box 6049
City	St. Mary's
State	ON
Country	CA
ZipCode	L6J 5L1
Title	Driver/Sales Workers
Company	St. Mary's Pest Control Ltd
Phone2	(555) 702-5520, #665
Email	a.tincher@oldrupa.com
Language	EN
MD5	BE9479B978EE0AE397083...
Number	INV8559825
Date	2013-03-15
DueDate	2013-04-05
Rep	REP8800122
SubTotal	125.96
TaxTotal	18.86
Total	144.82

Extract Step

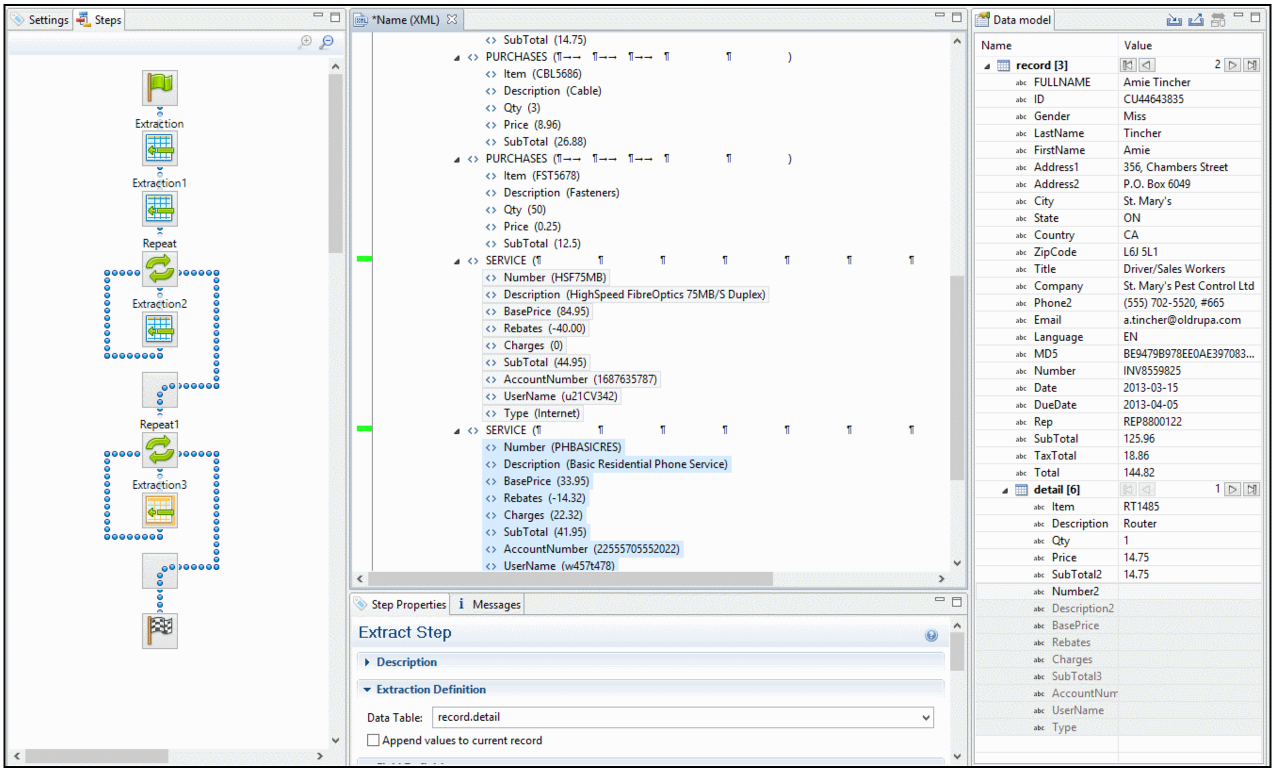
Description

Extraction Definition

Data Table: record

□ Append values to current record

4. Renaming Tables.



```
<> SubTotal (14.75)
<> PURCHASES (¶→→  ¶→→  ¶→→  ¶        ¶        )
    <> Item (CBL5686)
    <> Description (Cable)
    <> Qty (3)
    <> Price (8.96)
    <> SubTotal (26.88)
<> PURCHASES (¶→→  ¶→→  ¶→→  ¶        ¶        )
    <> Item (FST5678)
    <> Description (Fasteners)
    <> Qty (50)
    <> Price (0.25)
    <> SubTotal (12.5)
<> SERVICE (¶        ¶        ¶        ¶        ¶        ¶        ¶
    <> Number (HSF75MB)
    <> Description (HighSpeed FibreOptics 75MB/S Duplex)
    <> BasePrice (84.95)
    <> Rebates (-40.00)
    <> Charges (0)
    <> SubTotal (44.95)
    <> AccountNumber (1687635787)
    <> UserName (u21CV342)
    <> Type (Internet)
<> SERVICE (¶        ¶        ¶        ¶        ¶        ¶        ¶
    <> Number (PHBASICRES)
    <> Description (Basic Residential Phone Service)
    <> BasePrice (33.95)
    <> Rebates (-14.32)
    <> Charges (22.32)
    <> SubTotal (41.95)
    <> AccountNumber (22555705552022)
    <> UserName (w457t478)
```

Extract Step

- Description
- Extraction Definition

Data Table: record.detail

☐ Append values to current record

Name	Value
record [3]	2
FULLNAME	Amie Tincher
ID	CU44643835
Gender	Miss
LastName	Tincher
FirstName	Amie
Address1	356, Chambers Street
Address2	P.O. Box 6049
City	St. Mary's
State	ON
Country	CA
ZipCode	L6J 5L1
Title	Driver/Sales Workers
Company	St. Mary's Pest Control Ltd
Phone2	(555) 702-5520, #665
Email	a.tincher@oldrupa.com
Language	EN
MD5	BE9479B978EE0AE397083...
Number	INV8559825
Date	2013-03-15
DueDate	2013-04-05
Rep	REP8800122
SubTotal	125.96
TaxTotal	18.86
Total	144.82
detail [6]	1
Item	RT1485
Description	Router
Qty	1
Price	14.75
SubTotal2	14.75
Number2	
Description2	
BasePrice	
Rebates	
Charges	
SubTotal3	
AccountNum	
UserName	
Type	

Nested Tables

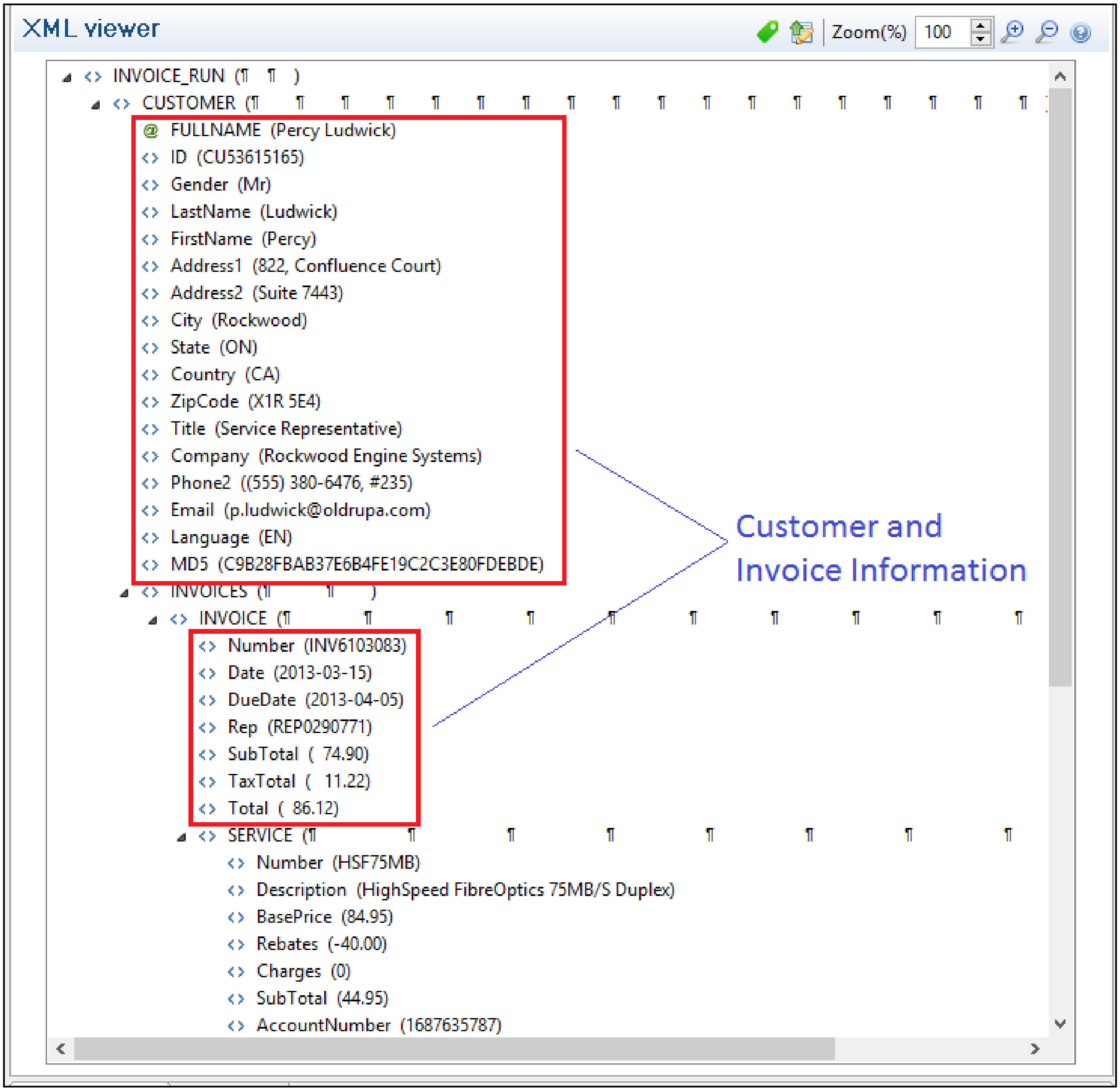
Nested detail tables are used to create transactional data that is relative to other data. An example of this would be an invoice for a multi-service provider. In this example, a first table contains services (Internet, Cable, Home Phone, Mobile), while one or more nested tables giving details for charges and rebates on each of those services.

Nested tables are created in a similar fashion to multiple detail tables, with the difference that the dot notation contains multiple levels. In the example above, tables could be called record.services , record.services.charges, record.services.details , where "charges" includes all service prices and rebates, and "details" includes extra items such as movie rentals or long distance calls.

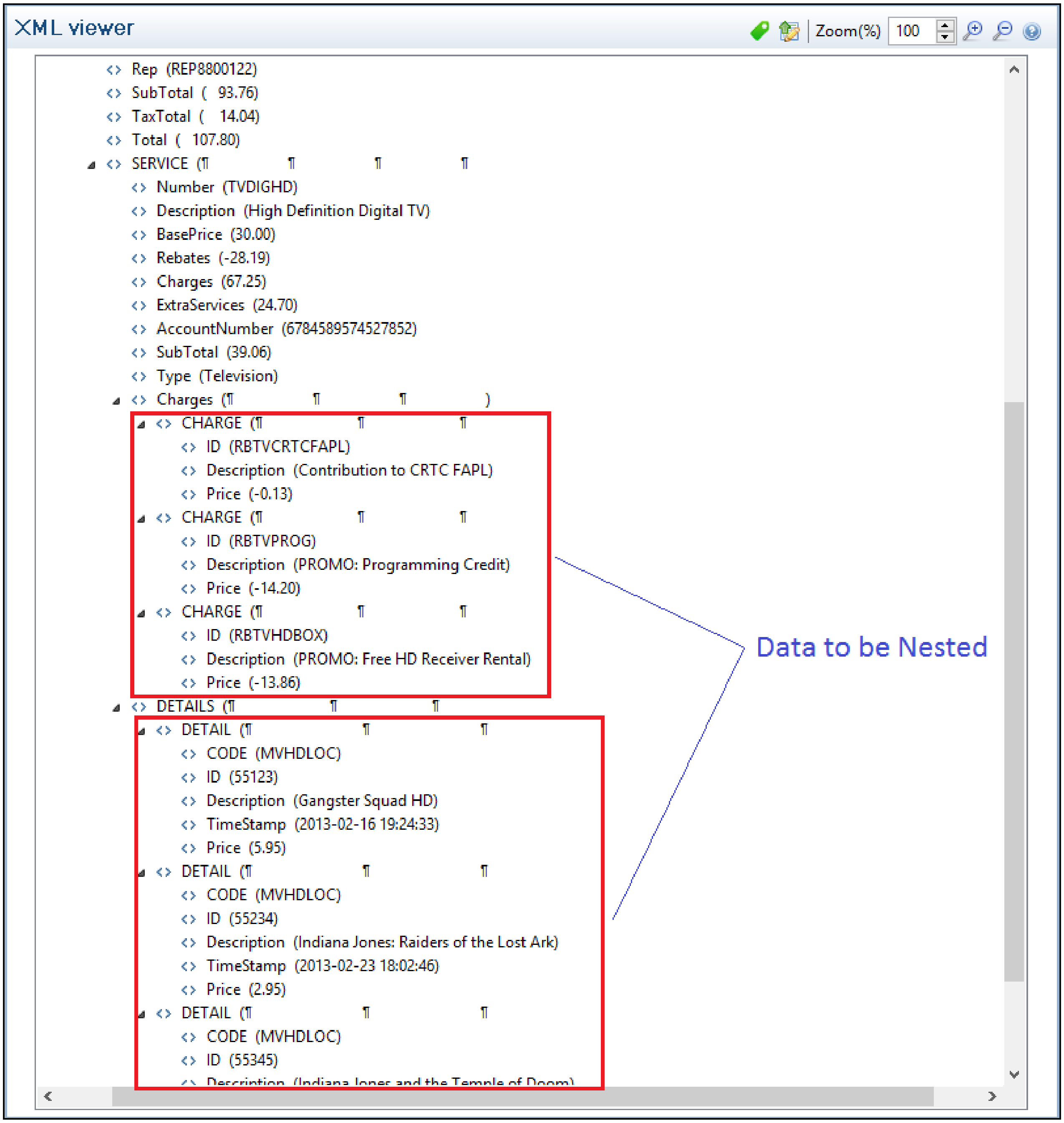
For the tables to be actually nested, the **Repeat** and its **Extract** step that extract the "charges" and "details" information must be located within the **Repeat** step that extracts data to "record.services". In such a setup, "record.services.charges" is a child table of "record.services", and one "charges" table is created for each row in the "services" table.

Example

1. Data Overview.



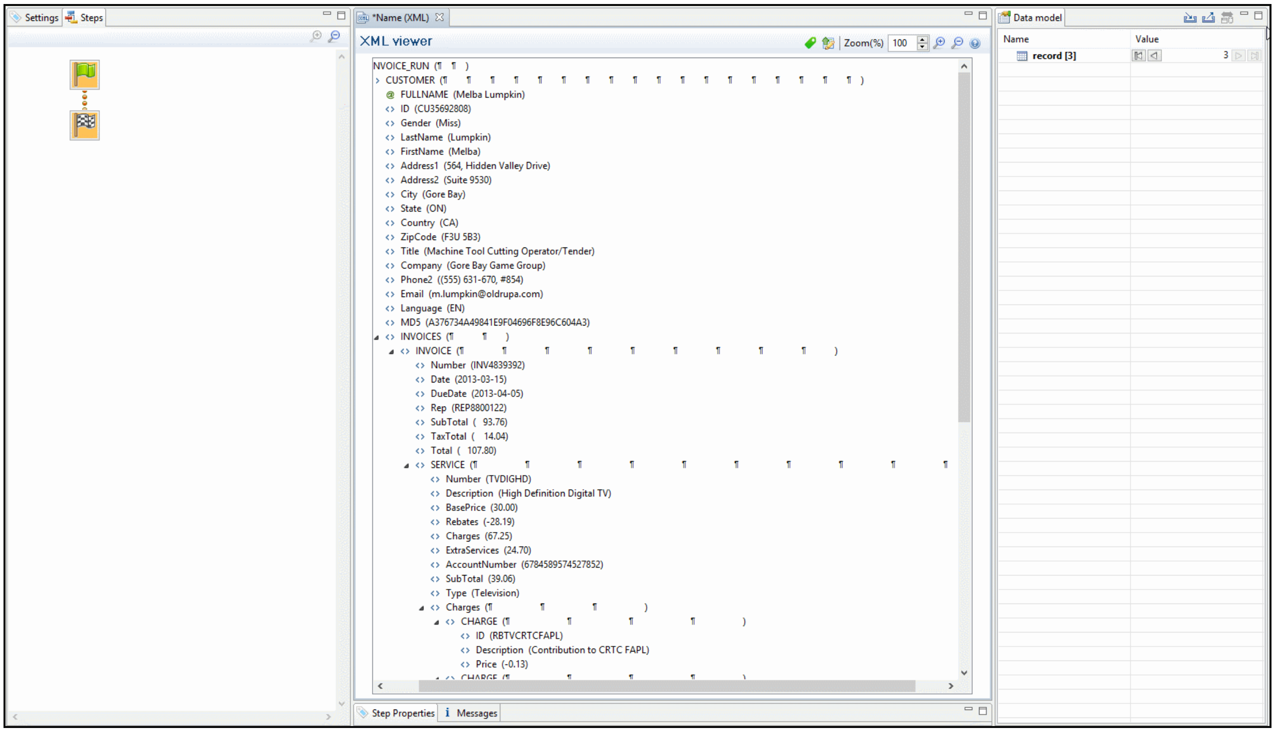
```
◢ <> INVOICE_RUN  (¶   ¶   )
   ◢ <> CUSTOMER  (¶     ¶     ¶     ¶     ¶     ¶     ¶     ¶     ¶     ¶     ¶     ¶     ¶     ¶     ¶     ¶     ¶     ¶
         @ FULLNAME  (Percy Ludwick)
         <> ID  (CU53615165)
         <> Gender  (Mr)
         <> LastName  (Ludwick)
         <> FirstName  (Percy)
         <> Address1  (822, Confluence Court)
         <> Address2  (Suite 7443)
         <> City  (Rockwood)
         <> State  (ON)
         <> Country  (CA)
         <> ZipCode  (X1R 5E4)
         <> Title  (Service Representative)
         <> Company  (Rockwood Engine Systems)
         <> Phone2  ((555) 380-6476, #235)
         <> Email  (p.ludwick@oldrupa.com)
         <> Language  (EN)
         <> MD5  (C9B28FBAB37E6B4FE19C2C3E80FDEBDE)
      ◢ <> INVOICES  (¶        ¶        )
         ◢ <> INVOICE  (¶        ¶        ¶        ¶        ¶        ¶        ¶        ¶        ¶        ¶
               <> Number  (INV6103083)
               <> Date  (2013-03-15)
               <> DueDate  (2013-04-05)
               <> Rep  (REP0290771)
               <> SubTotal  (  74.90)
               <> TaxTotal  (   11.22)
               <> Total  (  86.12)
            ◢ <> SERVICE  (¶        ¶        ¶        ¶        ¶        ¶        ¶        ¶
                  <> Number  (HSF75MB)
                  <> Description  (HighSpeed FibreOptics 75MB/S Duplex)
                  <> BasePrice  (84.95)
                  <> Rebates  (-40.00)
                  <> Charges  (0)
                  <> SubTotal  (44.95)
                  <> AccountNumber  (1687635787)
```



```
<> Rep (REP8800122)
<> SubTotal (  93.76)
<> TaxTotal (   14.04)
<> Total (  107.80)
◢ <> SERVICE (¶            ¶            ¶            ¶
    <> Number (TVDIGHD)
    <> Description (High Definition Digital TV)
    <> BasePrice (30.00)
    <> Rebates (-28.19)
    <> Charges (67.25)
    <> ExtraServices (24.70)
    <> AccountNumber (6784589574527852)
    <> SubTotal (39.06)
    <> Type (Television)
  ◢ <> Charges (¶            ¶            ¶            )
    ◢ <> CHARGE (¶            ¶            ¶
        <> ID (RBTVCRTCFAPL)
        <> Description (Contribution to CRTC FAPL)
        <> Price (-0.13)
    ◢ <> CHARGE (¶            ¶            ¶
        <> ID (RBTVPROG)
        <> Description (PROMO: Programming Credit)
        <> Price (-14.20)
    ◢ <> CHARGE (¶            ¶            ¶
        <> ID (RBTVHDBOX)
        <> Description (PROMO: Free HD Receiver Rental)
        <> Price (-13.86)
  ◢ <> DETAILS (¶            ¶            ¶
    ◢ <> DETAIL (¶            ¶            ¶
        <> CODE (MVHDLOC)
        <> ID (55123)
        <> Description (Gangster Squad HD)
        <> TimeStamp (2013-02-16 19:24:33)
        <> Price (5.95)
    ◢ <> DETAIL (¶            ¶            ¶
        <> CODE (MVHDLOC)
        <> ID (55234)
        <> Description (Indiana Jones: Raiders of the Lost Ark)
        <> TimeStamp (2013-02-23 18:02:46)
        <> Price (2.95)
    ◢ <> DETAIL (¶            ¶            ¶
        <> CODE (MVHDLOC)
        <> ID (55345)
        <> Description (Indiana Jones and the Temple of Doom)
```

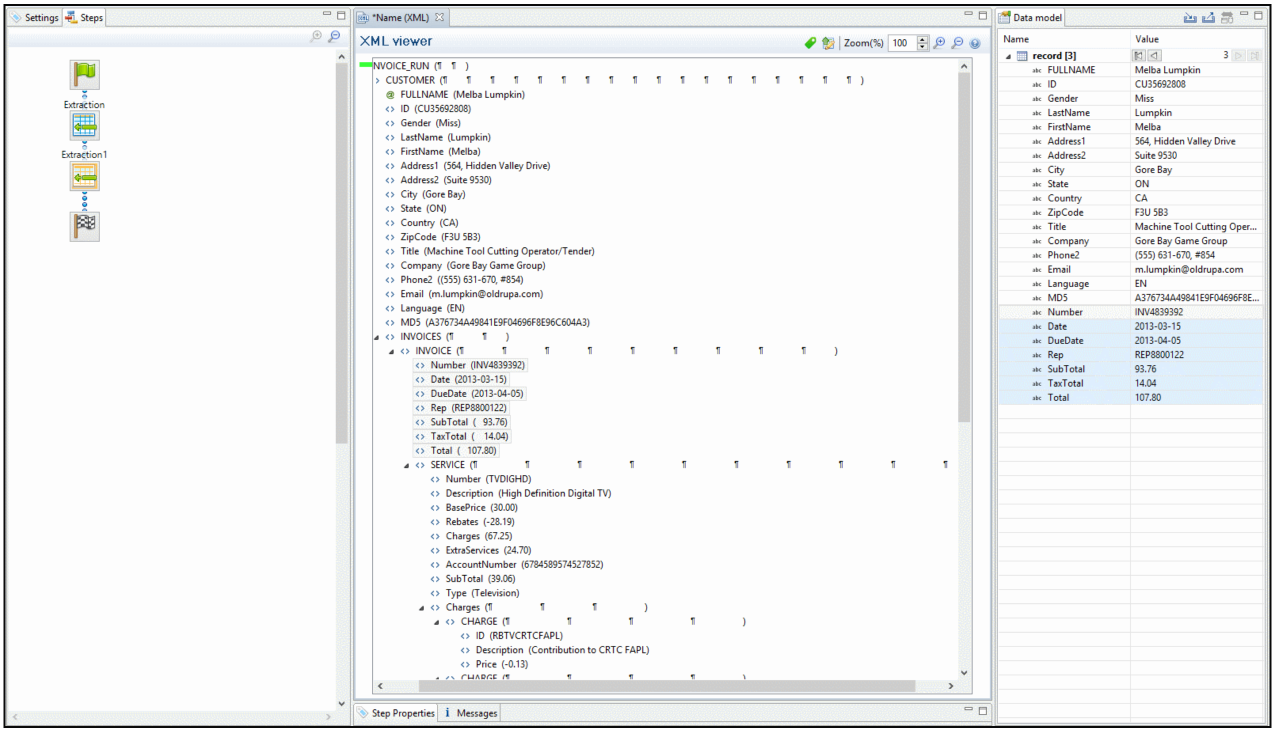
Data to be Nested

2. Extracting Customer and Invoice information.



```
NVOICE_RUN (¶  ¶  )
> CUSTOMER (¶   ¶   ¶   ¶   ¶   ¶   ¶   ¶   ¶   ¶   ¶   ¶   ¶   ¶   ¶   ¶   ¶   ¶ )
    @ FULLNAME (Melba Lumpkin)
    <> ID (CU35692808)
    <> Gender (Miss)
    <> LastName (Lumpkin)
    <> FirstName (Melba)
    <> Address1 (564, Hidden Valley Drive)
    <> Address2 (Suite 9530)
    <> City (Gore Bay)
    <> State (ON)
    <> Country (CA)
    <> ZipCode (F3U 5B3)
    <> Title (Machine Tool Cutting Operator/Tender)
    <> Company (Gore Bay Game Group)
    <> Phone2 ((555) 631-670, #854)
    <> Email (m.lumpkin@oldrupa.com)
    <> Language (EN)
    <> MD5 (A376734A49841E9F04696F8E96C604A3)
  ◢ <> INVOICES (¶      ¶    )
    ◢ <> INVOICE (¶      ¶      ¶      ¶      ¶      ¶      ¶      ¶      ¶    )
        <> Number (INV4839392)
        <> Date (2013-03-15)
        <> DueDate (2013-04-05)
        <> Rep (REP8800122)
        <> SubTotal (   93.76)
        <> TaxTotal (    14.04)
        <> Total (   107.80)
      ◢ <> SERVICE (¶       ¶       ¶       ¶       ¶       ¶       ¶       ¶       ¶       ¶
          <> Number (TVDIGHD)
          <> Description (High Definition Digital TV)
          <> BasePrice (30.00)
          <> Rebates (-28.19)
          <> Charges (67.25)
          <> ExtraServices (24.70)
          <> AccountNumber (6784589574527852)
          <> SubTotal (39.06)
          <> Type (Television)
        ◢ <> Charges (¶       ¶       ¶       )
          ◢ <> CHARGE (¶         ¶         ¶         ¶        )
              <> ID (RBTVCRTCFAPL)
              <> Description (Contribution to CRTC FAPL)
              <> Price (-0.13)
          ◢ <> CHARGE (¶         ¶         ¶         ¶        )
```

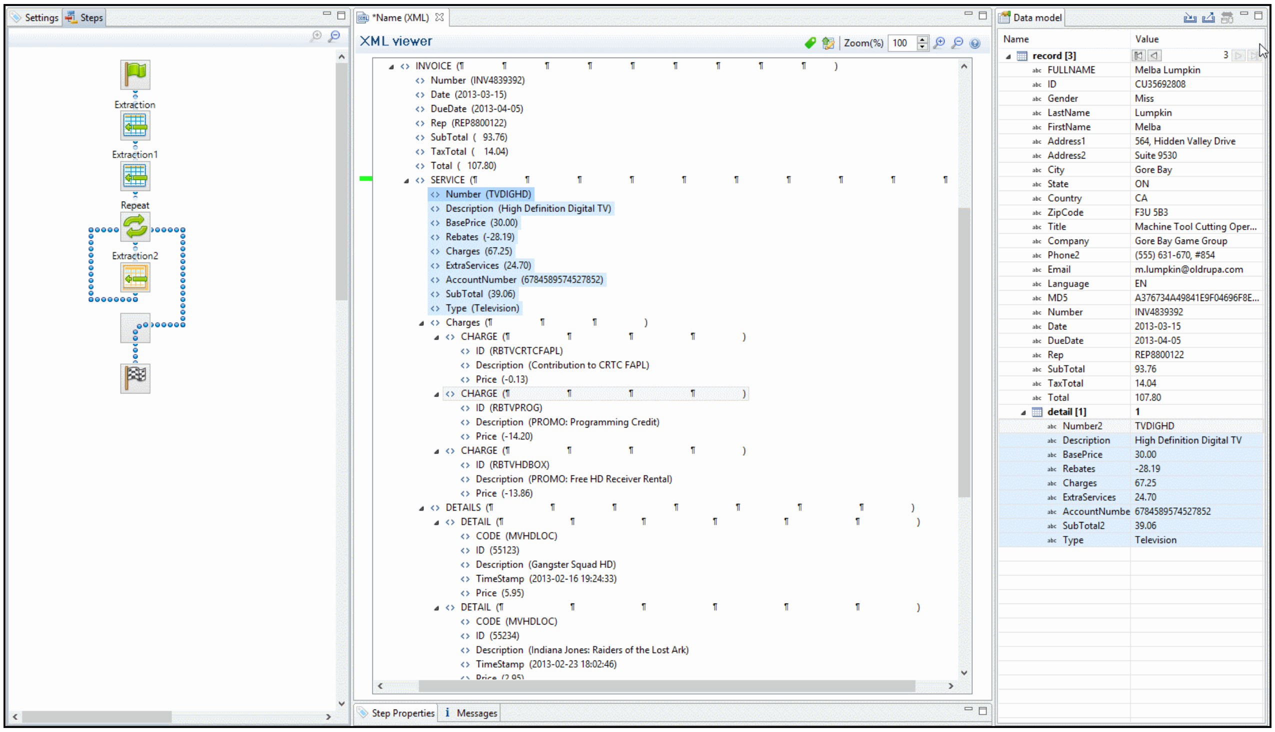
3. Extracting Transactional Data for the First Table.



```
NVOICE_RUN  (¶   ¶   )
> CUSTOMER  (¶   ¶   ¶   ¶   ¶   ¶   ¶   ¶   ¶   ¶   ¶   ¶   ¶   ¶   ¶   ¶   ¶   ¶   ¶   ¶  )
  @ FULLNAME  (Melba Lumpkin)
  <> ID  (CU35692808)
  <> Gender  (Miss)
  <> LastName  (Lumpkin)
  <> FirstName  (Melba)
  <> Address1  (564, Hidden Valley Drive)
  <> Address2  (Suite 9530)
  <> City  (Gore Bay)
  <> State  (ON)
  <> Country  (CA)
  <> ZipCode  (F3U 5B3)
  <> Title  (Machine Tool Cutting Operator/Tender)
  <> Company  (Gore Bay Game Group)
  <> Phone2  ((555) 631-670, #854)
  <> Email  (m.lumpkin@oldrupa.com)
  <> Language  (EN)
  <> MD5  (A376734A49841E9F04696F8E96C604A3)
◢ <> INVOICES  (¶        ¶     )
   ◢ <> INVOICE  (¶        ¶        ¶        ¶        ¶        ¶        ¶        ¶       )
        <> Number  (INV4839392)
        <> Date  (2013-03-15)
        <> DueDate  (2013-04-05)
        <> Rep  (REP8800122)
        <> SubTotal  (   93.76)
        <> TaxTotal  (   14.04)
        <> Total  (  107.80)
      ◢ <> SERVICE  (¶          ¶          ¶          ¶          ¶          ¶          ¶          ¶          ¶
           <> Number  (TVDIGHD)
           <> Description  (High Definition Digital TV)
           <> BasePrice  (30.00)
           <> Rebates  (-28.19)
           <> Charges  (67.25)
           <> ExtraServices  (24.70)
           <> AccountNumber  (6784589574527852)
           <> SubTotal  (39.06)
           <> Type  (Television)
         ◢ <> Charges  (¶          ¶          ¶          )
            ◢ <> CHARGE  (¶          ¶          ¶          )
                 <> ID  (RBTVCRTCFAPL)
                 <> Description  (Contribution to CRTC FAPL)
                 <> Price  (-0.13)
            ◢ <> CHARGE  (¶          ¶          ¶          )
```

Name	Value
record [3]	3
FULLNAME	Melba Lumpkin
ID	CU35692808
Gender	Miss
LastName	Lumpkin
FirstName	Melba
Address1	564, Hidden Valley Drive
Address2	Suite 9530
City	Gore Bay
State	ON
Country	CA
ZipCode	F3U 5B3
Title	Machine Tool Cutting Oper...
Company	Gore Bay Game Group
Phone2	(555) 631-670, #854
Email	m.lumpkin@oldrupa.com
Language	EN
MD5	A376734A49841E9F04696F8E...
Number	INV4839392
Date	2013-03-15
DueDate	2013-04-05
Rep	REP8800122
SubTotal	93.76
TaxTotal	14.04
Total	107.80

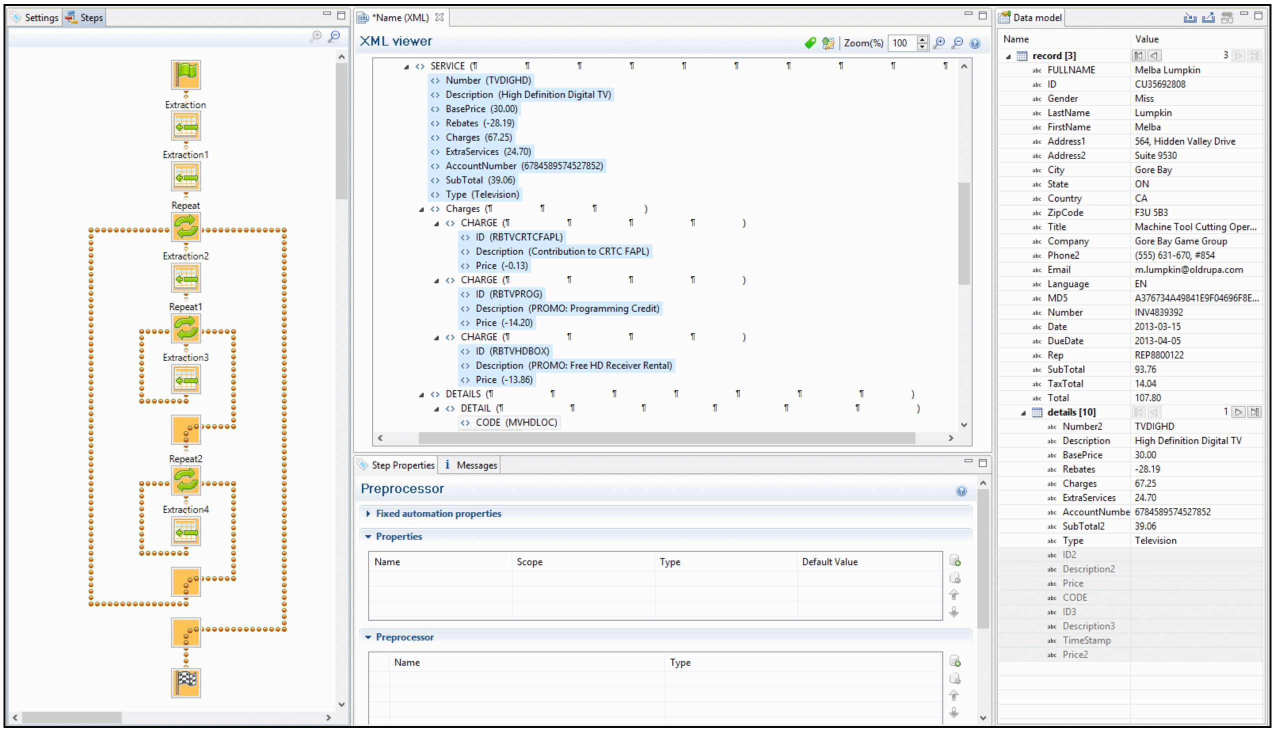
4. Extracting Transactional Data for Nested Tables.



```
◢ <> INVOICE (¶  ¶  ¶  ¶  ¶  ¶  ¶  ¶  ¶  )
    <> Number (INV4839392)
    <> Date (2013-03-15)
    <> DueDate (2013-04-05)
    <> Rep (REP8800122)
    <> SubTotal (  93.76)
    <> TaxTotal (   14.04)
    <> Total (  107.80)
  ◢ <> SERVICE (¶  ¶  ¶  ¶  ¶  ¶  ¶  ¶  ¶  ¶
      <> Number (TVDIGHD)
      <> Description (High Definition Digital TV)
      <> BasePrice (30.00)
      <> Rebates (-28.19)
      <> Charges (67.25)
      <> ExtraServices (24.70)
      <> AccountNumber (6784589574527852)
      <> SubTotal (39.06)
      <> Type (Television)
    ◢ <> Charges (¶  ¶  ¶  )
      ◢ <> CHARGE (¶  ¶  ¶  ¶  )
          <> ID (RBTVCRTCFAPL)
          <> Description (Contribution to CRTC FAPL)
          <> Price (-0.13)
      ◢ <> CHARGE (¶  ¶  ¶  ¶  )
          <> ID (RBTVPROG)
          <> Description (PROMO: Programming Credit)
          <> Price (-14.20)
      ◢ <> CHARGE (¶  ¶  ¶  ¶  )
          <> ID (RBTVHDBOX)
          <> Description (PROMO: Free HD Receiver Rental)
          <> Price (-13.86)
    ◢ <> DETAILS (¶  ¶  ¶  ¶  ¶  ¶  ¶  )
      ◢ <> DETAIL (¶  ¶  ¶  ¶  ¶  ¶  )
          <> CODE (MVHDLOC)
          <> ID (55123)
          <> Description (Gangster Squad HD)
          <> TimeStamp (2013-02-16 19:24:33)
          <> Price (5.95)
      ◢ <> DETAIL (¶  ¶  ¶  ¶  ¶  ¶  )
          <> CODE (MVHDLOC)
          <> ID (55234)
          <> Description (Indiana Jones: Raiders of the Lost Ark)
          <> TimeStamp (2013-02-23 18:02:46)
          <> Price (2.95)
```

Name	Value
record [3]	3
FULLNAME	Melba Lumpkin
ID	CU35692808
Gender	Miss
LastName	Lumpkin
FirstName	Melba
Address1	564, Hidden Valley Drive
Address2	Suite 9530
City	Gore Bay
State	ON
Country	CA
ZipCode	F3U 5B3
Title	Machine Tool Cutting Oper...
Company	Gore Bay Game Group
Phone2	(555) 631-670, #854
Email	m.lumpkin@oldrupa.com
Language	EN
MD5	A376734A49841E9F04696F8E...
Number	INV4839392
Date	2013-03-15
DueDate	2013-04-05
Rep	REP8800122
SubTotal	93.76
TaxTotal	14.04
Total	107.80
detail [1]	1
Number2	TVDIGHD
Description	High Definition Digital TV
BasePrice	30.00
Rebates	-28.19
Charges	67.25
ExtraServices	24.70
AccountNumbe	6784589574527852
SubTotal2	39.06
Type	Television

5. Renaming Tables.



```
◢ <> SERVICE (¶ ¶ ¶ ¶ ¶ ¶ ¶ ¶ ¶ ¶
    <> Number (TVDIGHD)
    <> Description (High Definition Digital TV)
    <> BasePrice (30.00)
    <> Rebates (-28.19)
    <> Charges (67.25)
    <> ExtraServices (24.70)
    <> AccountNumber (6784589574527852)
    <> SubTotal (39.06)
    <> Type (Television)
  ◢ <> Charges (¶ ¶ ¶ )
    ◢ <> CHARGE (¶ ¶ ¶ ¶ )
        <> ID (RBTVCRTCFAPL)
        <> Description (Contribution to CRTC FAPL)
        <> Price (-0.13)
    ◢ <> CHARGE (¶ ¶ ¶ ¶ )
        <> ID (RBTVPROG)
        <> Description (PROMO: Programming Credit)
        <> Price (-14.20)
    ◢ <> CHARGE (¶ ¶ ¶ ¶ )
        <> ID (RBTVHDBOX)
        <> Description (PROMO: Free HD Receiver Rental)
        <> Price (-13.86)
  ◢ <> DETAILS (¶ ¶ ¶ ¶ ¶ ¶ ¶ )
    ◢ <> DETAIL (¶ ¶ ¶ ¶ ¶ ¶ )
        <> CODE (MVHDLOC)
```

Preprocessor

Properties

Name	Scope	Type	Default Value

Preprocessor

Name	Type

Name	Value
◢ record [3]	3
FULLNAME	Melba Lumpkin
ID	CU35692808
Gender	Miss
LastName	Lumpkin
FirstName	Melba
Address1	564, Hidden Valley Drive
Address2	Suite 9530
City	Gore Bay
State	ON
Country	CA
ZipCode	F3U 5B3
Title	Machine Tool Cutting Oper...
Company	Gore Bay Game Group
Phone2	(555) 631-670, #854
Email	m.lumpkin@oldrupa.com
Language	EN
MD5	A376734A49841E9F04696F8E...
Number	INV4839392
Date	2013-03-15
DueDate	2013-04-05
Rep	REP8800122
SubTotal	93.76
TaxTotal	14.04
Total	107.80
◢ details [10]	1
Number2	TVDIGHD
Description	High Definition Digital TV
BasePrice	30.00
Rebates	-28.19
Charges	67.25
ExtraServices	24.70
AccountNumbe	6784589574527852
SubTotal2	39.06
Type	Television
ID2	
Description2	
Price	
CODE	
ID3	
Description3	
TimeStamp	
Price2	

Note

Creating nested tables is currently an advanced feature, and using these nested tables in the **Designer** module requires some amount of coding.

Note

For more information about operations that can be performed on tables, please refer to The Data Model Interface

The Data Viewer Interface

The Data Viewer displays the contents of the Data Sample that is currently active within the data mapping configuration. This data is never raw, meaning it has to be interpreted in some way to be displayed on the screen. For example, XML is displayed in a tree view rather than the XML code. Similarly, PDFs are displayed as pages.

When the Delimiter and Boundary options are set in the Settings Pane, the Data Viewer reflects those changes and can display more than one unit of the data sample (page, XML node, csv line, etc) depending on these options. The Data Viewer's display is also affected by the Preprocessor Step and displays the data after it has gone through all enabled preprocessors.

Window Controls

The following controls appear at the top of the Data Viewer:

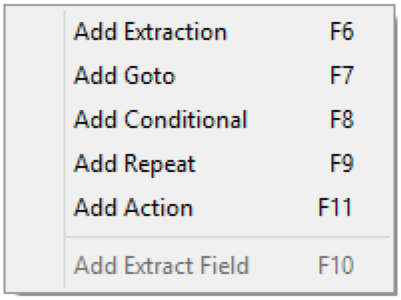
- **Font** (Text file only): Use the drop-down to change the font used to display text. Useful for double-byte data. It is recommended that monospace fonts be used.
- **Hide/Show line numbers** # (Text file only): Click to show or hide the line numbers on the left of the Data Viewer.
- **Hide/Show datamap** : Click to show or hide the icons to the left of the Data Viewer that displays how the steps affect the line.
- **Hide/Show extracted data** : Click to show or hide the extraction selections indicating that data is extracted. This simplifies making data selections in the same areas and is useful to display the original data.
- **Lock/Unlock extracted data** : Click to lock existing extraction selections so they cannot be moved or resized. This simplifies making data selections in the same area.
- **Zoom Level**: Use the arrows to adjust the zoom level, or type in the zoom percentage.
- **Zoom In (CTRL +)** : Click to zoom in by increments of 10%
- **Zoom Out (CTRL -)** : Click to zoom out by increments of 10%

Additional Keyboard Shortcuts for XML Files:

- **+** (while on an XML node with children): Expand the XML Node
- **-** (while on an XML node with children): Collapse the XML node, hiding all its children nodes.

Contextual Menu

You can access the contextual menu using a right-click anywhere inside the Viewer window:

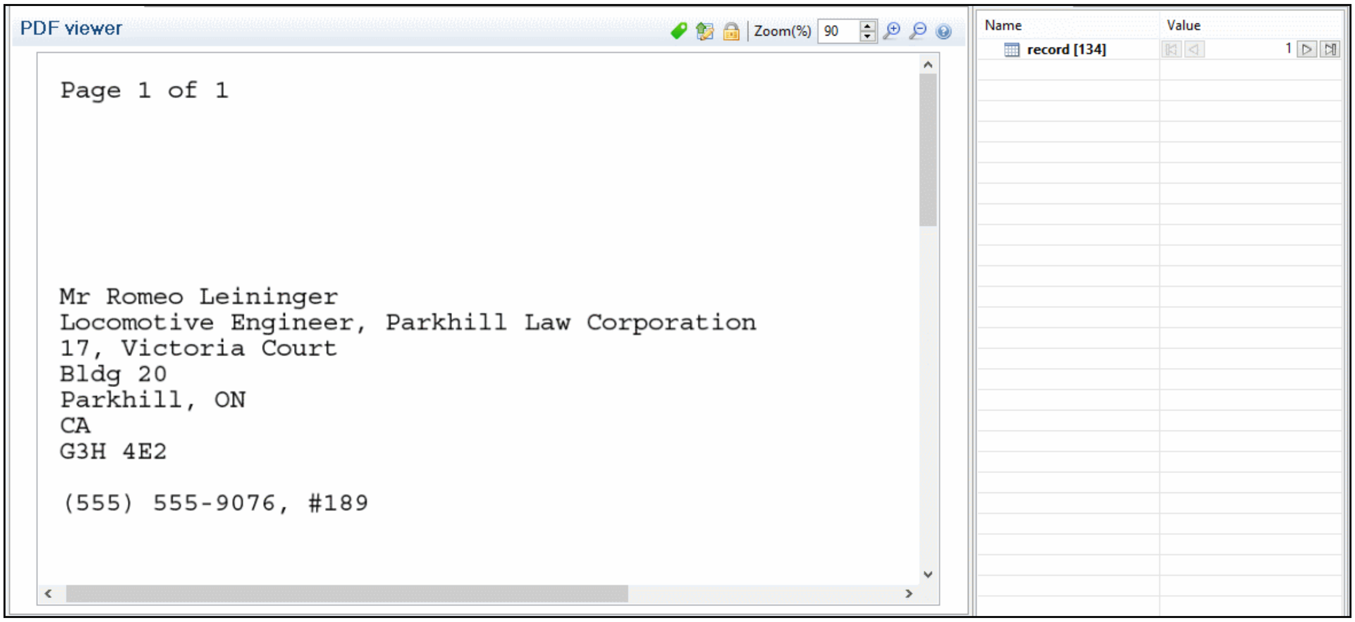


> **Note**
>
> The **Add Extract Field** item is available only after an **Extract** step has been added to the workflow.

For more information about the different steps that can be added to a Data Mapping workflow, please refer to Steps.

Operations

Drag & drops from the Viewer window to the Data Model pane can be performed to extract data.



```
Page 1 of 1

Mr Romeo Leininger
Locomotive Engineer, Parkhill Law Corporation
17, Victoria Court
Bldg 20
Parkhill, ON
CA
G3H 4E2

(555) 555-9076, #189
```

Messages Pane

The Messages pane is shared between the DataMapper and Designer modules and displays any warnings and errors from the data mapping configuration or template.

At the top of the Message pane are control 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.

- **Time**: The date and time when the error occurred.
- **Type**: Whether the entry is a warning or an error.
- **Source**: The source of the error. This indicates the name of the step as defined in its step properties.
- **Message**: The contents of the message, indicating the actual error.

Log Filter

The log filter determines what kind of events are show in the Messages Pane.

- **Event Types** group:
  - **OK**: Uncheck to hide OK-level entries.
  - **Information**: Uncheck to hide information-level entries.
  - **Warning**: Uncheck to hide any warnings.
  - **Error**: Uncheck to hide any critical errors.
- **Limit visible events to**: Enter the maximum number of events to show in the Messages Pane. Default is 50.

The Settings Pane

The **Settings** Pane is used to configure the Data Sample. The two main components that are configured are Delimiters and Boundaries. The available options depend on the type of data sample that is loaded. For more information about Delimiters and Boundaries, see Configuring The Data Source (Settings).

The Input Data Section

For a CSV File

- **Field separator**: Defines what character separates each fields in the file.
- **Text delimiter**: Defines what character surrounds text fields in the file, preventing the **Field separator** from being interpreted within those text delimiters.
- **Comment delimiter**: Defines what character starts a comment line.
- **Encoding**: Defines what encoding is used to read the Data Source (US-ASCII, ISO-8859-1, UTF-8, UTF-16, UTF-16BE or UTF-16LE).
- **Lines to skip**: Defines a number of lines in the CSV that will be skipped and not be used as Source Records.
- **Set tabs as a field separator**: Overwrites the **Field separator** option and sets the Tab character instead for tab-delimited files.
- **First row contains field names**: Uses the first line of the CSV as headers, which automatically names all extracted fields.
- **Ignore unparseable lines**: Ignores any line that does not correspond to the settings above.

For a PDF File

> **Note**
>
> PDF Files have a natural, static delimiter in the form of Pages, so the options here are interpretation settings for text in the PDF file. Each value represents a fraction of the average font size of text in a data selection, meaning ".3" represents 30% of the height or width.

- **Word spacing**: Determines the spacing between words. As PDF text spacing is somehow done via positioning instead of actual text spaces, position of text is what is used to find new words. This option determines what percentage of the average width of a single character needs to be empty to consider a new word has started. Default value is .3 , meaning a space is assumed if a blank area of 30% of width of the average character in the font.
- **Line spacing**: Determines the spacing between lines of text. The default value is 1, meaning the space between lines must be equal to at least the height of the average character height.

- **Paragraph spacing**: Determines the spacing between paragraphs. The default value is 1.5, meaning the space between paragraphs must be equal to at least 1.5 times the height of the average character height to start a new paragraph.
- **Magic number**: Determines the tolerance factor for all the above values. The tolerance is meant to avoid rounding errors. If two values are more than 70% away from each other they are considered distinct, otherwise they are the same. For example, if two characters have a space of exactly one times the width of the average character, we actually consider any space of between ".7" and "1.43" this average width equals to one space. A space of 1.44 is considered to be 2 spaces.
- **PDF file color space**: Determines if the PDF if displayed in Color or Monochrome in the Data Viewer. Monochrome display is faster in the Data Viewer, but this has no influence on actual data extraction or the data mapping performance.

For Databases

The following settings apply to any database and ODBC Data Sample.

- **Connection String**: Displays the connection string used to access the Data Source.
- **Table**: Displays the tables and stored procedures available in the database. The selected table is the one the data is extracted from.
- **Encoding**: Defines what encoding is used to read the Data Source (US-ASCII, ISO-8859-1, UTF-8, UTF-16, UTF-16BE or UTF-16LE).
- **Browse** button : Opens the **Edit Database configuration** dialog, which can replace the existing database data source with a new one. This is the same as using the **Replace** feature in the Data Samples window.
- **Custom SQL** button : Click to open the SQL Query Designer and type in a custom SQL query.

For a Text File

- **Encoding**: Defines what encoding is used to read the Data Source (US-ASCII, ISO-8859-1, UTF-8, UTF-16, UTF-16BE or UTF-16LE).
- **Add/Remove characters**: Defines the number of characters to add to, or remove from, the head of the data stream. The spin buttons can also increment or decrement the value. Positive values add blank characters while negative values remove characters.
- **Add/Remove lines**: Defines the number of lines to add to, or remove from, the head of the data stream. The spin buttons can also increment or decrement the value. Positive values add blank lines while negative values remove lines.
- **Maximum line length**: Defines the number of columns in a data page. The spin buttons can also increment or decrement the value. The maximum value for this option is 65,535 characters. The default value is 80 characters. You should tune this value to the longest

line in your input data. Setting a maximum data line length that greatly exceeds the length of the longest line in your input data may increase execution time.

- **Page delimiter type**: Defines the delimiter between each page of data. Multiple such pages can be part of a Source Record, as defined by the **Boundaries**.
    - **On lines**: Triggers a new page in the Data Sample after a static number of lines (called Lines per Page), or using a Form Feed character.
    - **On text**: Triggers a new page in the Data Sample when a specific string is found in a certain location.
        - **Word to find**: Compares the text value with the value in the Source Record.
        - **Match case**: Activates case-sensitive text comparison.
        - **Location**: Choose **Selected area** or **Entire width** to use the value of the current data selection as the text value.
        - **Left/Right**: Use the spin buttons to set the start and stop columns to the current data selection (**Selected area**) in the Source Record.
        - **Lines before/after**: Defines the delimiter a certain number of lines before or after the current line. This is useful if the text triggering the delimiter is not on the first line of each page.

For a XML File

Delimiters for XML files are related to XML Elements, or locations in its hierarchy.

- **Use root element**: Locks the **XML Elements** option to the top-level element. If there is only one top-level element, there will only be one record before the Boundaries are set.
- **XML elements**: Displays a list containing all the elements in the XML file. Selecting an element causes a new page of data to be created every time an instance of this element is encountered.

The Boundaries Section

When the Data Source is received by the DataMapper, it has no boundaries to tell the DataMapper if it contains different records or where each of those records begins and ends. That's because boundaries are not actual bits of data (like a character or a field would be); boundaries are a logical structure that exists outside the Data Source (note that some formats like PDF/VT actually include structured information, but those are the exception rather than the norm). Boundaries are therefore a form of metadata. You could very well use the same exacted data with a different boundary structure, in order to extract different information. Think, for instance, of an Invoice Run stored in a PDF. You could very well be using a structure where each invoice is a single record or you could group all invoices for one customer into a single record. So the boundaries for each record can be completely dependent on how you want to use the data.

With no actual boundary markers inside the data, we have to figure out a way to identify specific locations in the input stream and mark those locations as record boundaries. Fortunately for us, it turns out that every single file format possesses intrinsic, natural delimiters that are used to identify chunks of related data. These delimiters are key in helping us identify boundaries, so it's important to understand what they are as well as when and why they occur in the Data Source.

Let's start with a seemingly arbitrary assumption: a boundary can only occur on a natural delimiter. That is to say, a record boundary never occurs between delimiters, it only occurs on a delimiter. The actual information we need to determine whether a delimiter is a candidate for being a boundary is very likely to be found between delimiters.

To do that, we'll take a closer look at the various input data types.

For a CSV or Database File

The natural delimiter for a CSV file is a data record, or to put it more visually, each line in a spreadsheet or in a SQL data grid is a delimiter. Several such delimiters can be included in a record, but you would never expect to find the end of one particular record right in the middle of one of those lines on the grid. So here, the record occurs with a new line in the grid, but not on each new line.

- **Record limit**: Defines how many Source Records are displayed in the Data Viewer. This does not affect output production, as generating output ignores this option. To disable the limit, use the value "0".
- **Line limit**: Defines the limit of detail lines in any detail table. This is useful for files with a high number of detail lines, which in the DataMapper interface can slow down things. This does not affect output production, as generating output ignores this option. To disable the limit, use the value "0".
- **Trigger**: Defines the type of rule that control when a boundary is created, establishing a new record in the Data Sample (called a Source Record).
    - **Record(s) per page**: Defines a set number of lines in the file that go in each Source Record.
        - **Records**: The number of records to show in each Source Records.
    - **On change**: Defines a new Source Record when a specific field (**Field name**) has a new value.

    - **Field name**: Displays the fields in the top line. The selected value determines new boundaries.
  - **On script:** Defines the boundaries using a custom user-defined JavaScript. For more information see Boundaries Using javaScript.
  - **On field value**: Defines the boundary when the contents of a specific field value.
    - **Field name**: Displays the fields in the top line. The selected value is compared with the **Expression** below to create a new boundary.
    - **Expression**: Enter the value or **Regular Expression** that triggers a new boundary when it is the field value.
    - **Use Regular Expression**: Treats the **Expression** as a regular expression instead of static text. For more information on using **Regular Expressions** (regex), see the Regular-Expressions.info Tutorial.

For a PDF File

For a PDF file, the natural delimiter is a Page. A record boundary always occurs on a new page, never in the middle of a page. Each record may very well include one or more pages (i.e. delimiters).

> **Note**
>
> While a record boundary always occurs on a new page, the opposite is not true: a new page is not always a record boundary.

- **Record limit**: Defines how many Source Records are displayed in the Data Viewer. To disable the limit, use the value "0".
- **Trigger**: Defines the type of rule that defines when a boundary is created, establishing a new record in the Data Sample (called a Source Record).
  - **On page**: Defines a boundary on a static number of pages.
    - **Number of pages**: Defines how many pages are in each Source Record.
  - **On text**: Defines a boundary on a specific text comparison in the Source Record.
    - **Start coordinates (x,y)**: Defines the left and top coordinates of the data selection to compare with the text value.
    - **Stop coordinates (x,y)**: Defines the right and bottom coordinates.
    - **Use Selection**: Select an area in the Data Viewer and click the **Use selection** button to set the start and stop coordinates to the current data selection in the Source Record.

        - **Times condition found**: When the boundaries are based on the presence of specific text, you can specify after how many instances of this text the boundary can be effectively defined. For example, if a string is always found on the first and on the last page of a document, you could specify a number of occurrences of 2. In this way, no need to inspect other items for whether it is on the first page or the last page. you know you have found the string two times, which is enough to fix the boundary.
        - **Pages before/after**: Defines the boundary a certain number of pages before or after the current page. This is useful if the text triggering the document is not located on the first page of each Source Record.
        - **Operator**: Selects the type of comparison that is done (for example, "contains").
        - **Word to find**: Compares the text value with the value in the Source Record.
        - **Match case**: Makes the text comparison case-sensitive.

For a Text File

For a Text file, the natural delimiter is also a Page, but contrary to PDF, the Page delimiter can either be explicit (say, when a Form Feed character is encountered in the Data Source) or implicit (when a certain number of lines has been reached - usually around 66). Once more, you wouldn't find the end of a record in the middle of a line. Note also that with this format, it's possible to set the DataMapper's Input Data settings to 1 line per page. That essentially allows you to set the natural delimiter on each and every line in the file.

- **Record limit**: Defines how many Source Records are displayed in the Data Viewer. To disable the limit, use the value "0".
- **Trigger**: Defines the type of rule that defines when a boundary is created, establishing a new record in the Data Sample (called a Source Record).
    - **On delimiter:** Defines a boundary on a static number of pages.
        - **Occurrences**: The number of times that the delimiter is encountered before fixing the boundary. For example, if you know that your documents always have four pages delimited by the FF character, you can set the boundaries whenever you counted four delimiters.
    - **On text**: Defines a boundary on a specific text comparison in the Source Record.
        - **Location**:
            - **Selected area**:
                - **Select the area** button: Uses the value of the current data selection as the text value.

            - **Left/Right**: Defines the location, in the row, where to find the text value.
            - **Top/Bottom**: Defines the start and end row of the data selection to compare with the text value.
        - **Entire width**: Ignores the column values and compares using the whole line.
        - **Entire height**: Ignores the row values and compares using the whole column.
        - **Entire page**: Compares the text value on the whole page. Only available with *contains*, *not contains*, *is empty* and *is not empty* operators.
    - **Times condition found**: When the boundaries are based on the presence of specific text, you can specify after how many instances of this text the boundary can be effectively defined. For example, if a string is always found on the first and on the last page of a document, you could specify a number of occurrences of 2. In this way, no need to inspect other items for whether it is on the first page or the last page. you know you have found the string two times, which is enough to fix the boundary.
    - **Delimiters before/after**: Defines the boundary a certain number of pages before or after the current page. This is useful if the text triggering the document is not located on the first page of each Source Record.
    - **Operator**: Selects the type of comparison that is done (for example, "contains").
    - **Word to find**: Compares the text value with the value in the Source Record.
    - **Match case**: Makes the text comparison case-sensitive.
- **On script:** Defines the boundaries using a custom user-defined JavaScript. For more information see Boundaries Using javaScript.

For a XML File

For a XML file, the natural delimiter is an XML Node.

- **Record limit**: Defines how many Source Records are displayed in the Data Viewer. To disable the limit, use the value "0".
- **Trigger**: Defines the type of rule that defines when a boundary is created, establishing a new record in the Data Sample (called a Source Record).
    - **On Element**: Defines a new Source Record on each new instance of the XML level selected in the **XML elements**.

    - **Occurrences**: The number of times that the delimiter is encountered before fixing the boundary. For example, if you know that your documents always have four pages delimited by the FF character, you can set the boundaries whenever you counted four delimiters.
- **On Change**: Defines a new Source Record when a specific field under the XML level has a new value.
    - **Field**: Displays the fields that are under the XML level. The selected one's value determines new boundaries.

The Data Samples Section

Instead of buttons listed below, you can also right-click to bring up the context menu which offers the same options.

- **Add** : Adds a new Data Sample from an external Data Source. The new Data Sample will need to be of the same data type as the current one. For example, you can only add PDF files to a PDF data mapping configuration. In version 1.3 and higher, multiple files can be added simultaneously through the Add dialog.
- **Delete** : Removes the current Data Sample from the data mapping configuration.
- **Replace** : Opens a new Data Sample and replaces it with the contents of a different data source.
- **Reload** : Reloads the currently selected Data Sample and any changes that have been made to it.
- **Set as Active** : Activates the selected Data Sample. The active data sample is shown in the Data Viewer after it has gone through the **Preprocessor** step as well as the **Input Data** and **Boundary** settings.

The External JS Libraries Section

Right-clicking in the box brings up a control menu, also available through the buttons on the right:

- **Add** : Adds a new external library. Opens the standard **Open** dialog to browse and open the .js file.
- **Delete** : Removes the currently selected library from the data mapping configuration.
- **Replace** : Opens a new library and replaces it with the contents of a different js file.
- **Reload** : Reloads the currently selected library and any changes that have been made to it.

Default Data Format

Default Format Settings can also be defined at the Datamapper configuration level (see Datamapper Default Data Format for more information).

SQL Query Designer

The SQL Query Designer is used to design a custom SQL query to pull information from a database.

- **Tables**: Lists all tables and stored queries in the database.
- **Custom Query**: Displays the query that retries information from a database. Each database type has their own version of the SQL query language. To learn how to build your own query, please refer to your database's user manual.
- **Test Query** button ✓: Click to test the custom query to ensure it will retrieve the appropriate information.
- **Results**: Displays the result of the SQL query when clicking on **Test Query**.

The Steps Pane Interface

The Steps Pane displays each step of the workflow of which the primary goal is to extract data from the data sample.

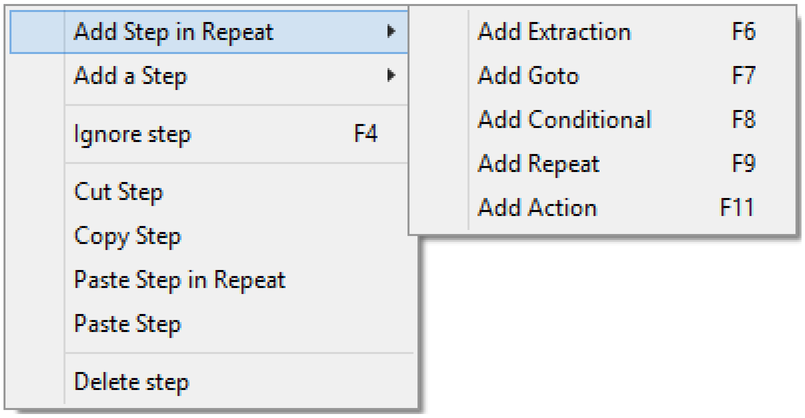
Window Controls

The following controls appear at the top of the Steps pane:

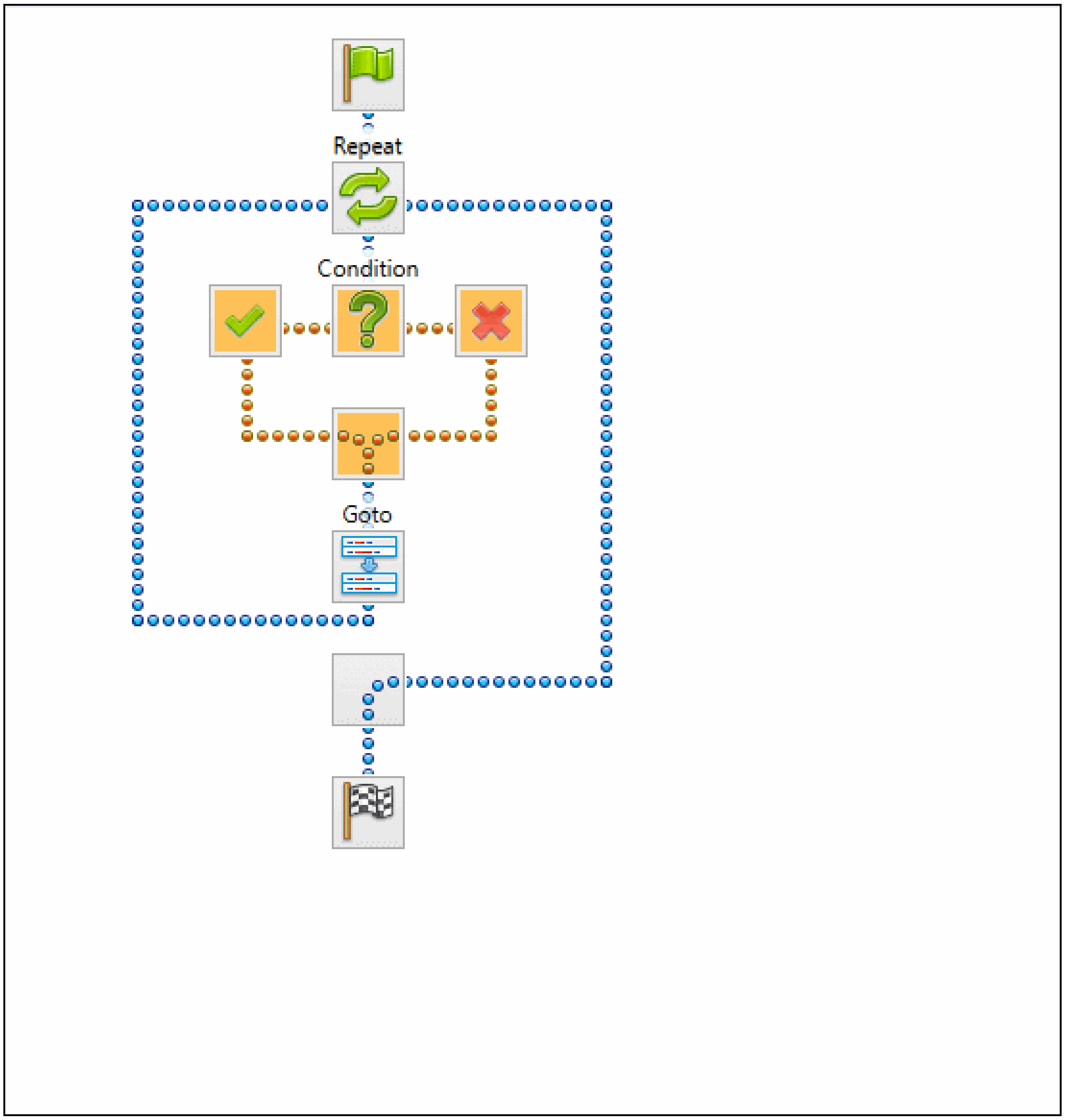
- **Zoom In (CTRL +)**⊕: Click to zoom in by increments of 10%
- **Zoom Out (CTRL -)**⊖: Click to zoom out by increments of 10%

Contextual Menu

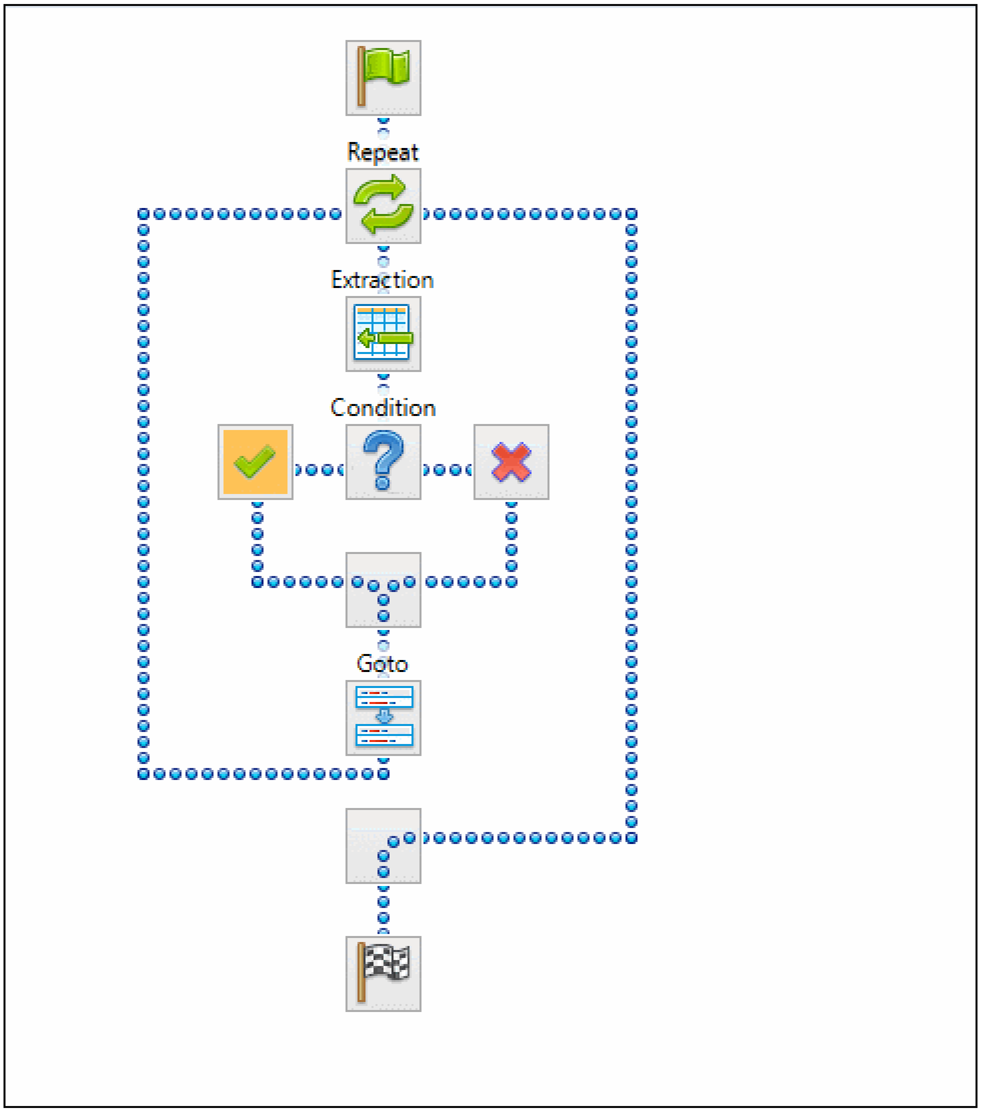
You can access the contextual menu using a right-click anywhere inside the **Steps** pane:



- **Add a Step**: Adds a step to the process. Right click anywhere in the **Steps** pane and select **Add a Step** from the contextual menu. For more information about the steps and how to use them, please refer to Steps. More options are available when a **Repeat** or a **Condition** step is selected:
  - **Add Step in Repeat**: To add a step into the **Repeat** loop, right-click on the step and select **Add a Step in Repeat**.



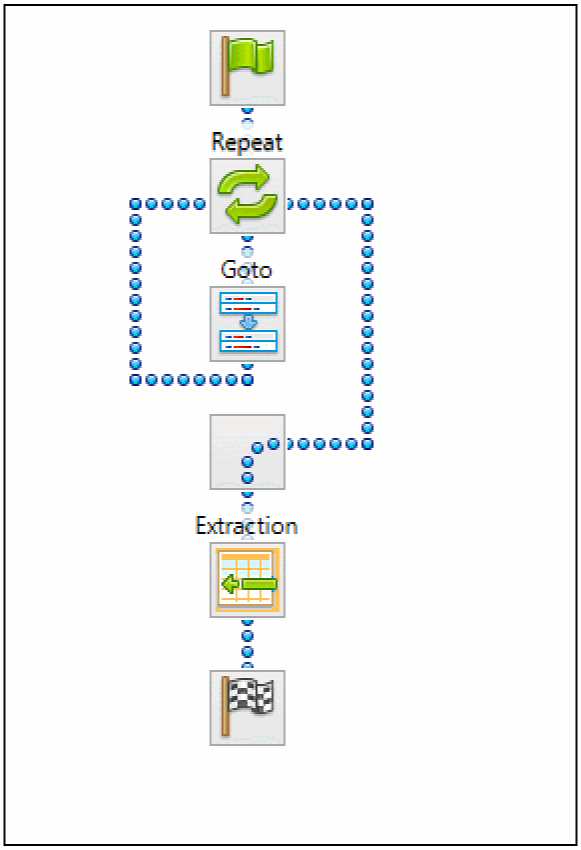
- **Add Step in True**: To add a step under the true branch of a **condition** step, right-click on the condition and select **Add a Step in True**.



  - **Add Step in False**: To add a step under the false branch of a **condition** step, right-click on the condition and select **Add a Step in False**.
- **Ignore Step**: Click to set the step to be ignored (aka disabled). Disabled steps are grayed and do not run when in DataMapper and do not execute when the data mapping configuration is executed in Workflow. However, they can still be modified normally.
- **Moving**: To move a step, right-click on it and select **Cut Step** or use the ✂ button in the **Toolbar**. If the step is **Repeat** or **Condition**, all steps under it will also be placed in the clipboard. If there is already a step in the clipboard, it will be overwritten. To place the step at its destination, right-click the step in the position before the desired location and click **Paste Step**, or use the 📋 button in the toolbar.

Note

You can also use a drag & drop to move steps.



- **Delete Step**: To remove a step, right-click on it and select **Delete step** from the contextual menu or use the ✖ button in the **Toolbar**. If the step to be deleted is **Repeat** or **Condition**, all steps under it will also be deleted.

Warning

Currently there is no **Undo** and **Redo** feature in the DataMapper module. Deleting a step is **permanent**!

- **Copy Step**: To copy a step, right-click on it and select **Copy Step** or use the button in the Toolbar. If the step is **Repeat** or **Condition**, all steps under it will also be placed in the clipboard. If there is already a step in the clipboard, it will be overwritten. To place a copy of the step at its destination, right-click the step in the position before the desired location and select **Paste Step**, or use the button in the **Toolbar**.
- **Viewing step details**: To see details for the step, click on the step to show its properties in the Step Properties pane. Hovering over the task shows a tooltip that displays some of the details as well as the property name and comment.

Step Properties

The **Step Properties** pane displays the properties for any step selected in the Steps Pane.

> **Note**
>
> Step properties may also depend on the Data Sample's file type.

Preprocessor Step

Extract Step

Action Step

Repeat Step

Condition Step

Goto Step

Postprocessor Step

Preprocessor Step Properties

The **Preprocessor** step does not run for every Source Record in the Data Sample. It runs once, at the beginning of the **Steps**, before anything else.

Fixed Automation Properties

The **Fixed automation properties** section lists all the fixed properties available from the PReS Workflow automation module. These properties are equivalent to data available within the PReS Workflow process. For each property, the following is available:

- **Name**: A read-only field displaying the name of the property.
- **Scope**: A read-only field indicating that the scope of the property is **Automation**.
- **Type**: A read-only field indicating the data type for each property.
- **Default Value**: Enter a default value for the property. This value is overwritten by the actual value coming from PReS Workflow when the data mapping configuration is run using the **Execute Data Mapping** task.

There are currently the following automation properties available:

- **JobInfoX**: These properties are the equivalent of the JobInfo values available in the PReS Workflow process. They can be set using the Set Job Info and Variables task. To access these properties inside of any JavaScript code within the Data Mapping Configuration, use the automation.jobInfos.JobInfoX (where X is the job info number, from 0 to 9).
- **OriginalFilename**: This property contains the original file name that was captured by the PReS Workflow process and is equivalent to the %o variable in the process. To access these property inside of any JavaScript code within the Data Mapping Configuration, use automation.properties.OriginalFilename.
- **ProcessName**: This property contains the name of the process that is currently executing the data mapping configuration and is equivalent to the %w variable in the process. To access this property inside of any JavaScript code within the Data Mapping Configuration, use automation.properties.ProcessName.
- **TaskIndex**: This property contains the index (position) of the task inside the process that is currently executing the data mapping configuration but it has no equivalent in PReS Workflow. To access this property inside of any JavaScript code within the Data Mapping Configuration, use automation.properties.ProcessName.

Properties

The **Properties** section is used to create specific properties that are used throughout the workflow. Properties can be accessed through some of the interface elements such as the **Condition** and **Repeat** step properties, or through the JavaScript API.

> **Note**
>
> Properties are evaluated in the order they are placed in the list, so properties can use the values of previously defined properties in their expression.

- **Name**: The name of the property used to refer to its value.
- **Scope**: What this property applies to:
    - **Entire Data**: These properties are static properties that cannot be changed once they have been set, in other words they are **Global constants**.
    - **Each Record**: These properties are evaluated and set at the beginning of each Source Record. These properties can be modified once they have been set, but are always reset at the beginning of each Source Record.
    - **Automation variable**: These properties initialize variables coming from the PReS Workflow automation tool. The name of the property needs to be the same as the variable name in Workflow, and they can be either a Local variable or a Global variable. For either one, only the actual name is to be used, so for %{MyLocalVar} use only MyLocalVar , and for %{global.MyGlobalVar} use MyGlobalVar. If a global and a local variable have the same name (%{myvar} and %{global.myvar}), the local variable's value is used and the global one is ignored. To access a workflow variable inside of any JavaScript code within the Data Mapping Configuration, use automation.variables.variablename
- **Type**: The data type of the property. For more information see Data Types.
- **Default Value**: The initial value of the property. This is a JavaScript expression. See JavaScript API.

Technical

Entire **Data Properties** are evaluated before anything else, such as **Preprocessors**, **Delimiters** and **Boundaries** in the Settings Pane. This means these properties cannot read information from the Data Sample or from any records. They are mostly useful for static information such as folder locations or server addresses.

Preprocessor

The **Preprocessor** section defines what preprocessors run on the Data Sample before it is sent to the extraction. Preprocessors modify the Data Sample in many ways, and each **Preprocessor** runs in turn, using the result of the previous one as an input.

- **Name**: The name to identify the **Preprocessor**.
- **Type**: The type of **Preprocessor**. Currently there is only one type available.

Preprocessor definition

- **Expression**: The JavaScript expression that will run on the Data Sample. See JavaScript API.

Description
This section is collapsed by default in the interface in order to give more screen space to important parts.

Name: The name of the step. This name will be displayed on top of the step's icon in the **Steps** pane.

Comments: The text entered here gives more details on the step and will be displayed in the tooltip appearing when hovering over the step in the **Steps** pane.

Extract Step Properties

The **Extract** step takes information from the Data Sample and places it in the Extracted Record within the Record Set.

Description
This section is collapsed by default in the interface in order to give more screen space to important parts.

Name: The name of the step. This name will be displayed on top of the step's icon in the **Steps** pane.

Comments: The text entered here gives more details on the step and will be displayed in the tooltip appearing when hovering over the step in the **Steps** pane.

Extraction Definition

- **Data Table**: Defines where the data will be placed in the **Extracted Record**. The root table is record, any other table becomes a detail table. For more information see detail tables.
- **Append values to current record**: When the **Extract Step** is within a loop, check this to ensure that the extraction will be done in the same **detail line** as any previous extractions within the same loop. This ensures that, if multiple extracts are present, only one detail line is created.

Field Definition

This section explains how the currently selected **Extract Field** in the **Field List** is defined.

Text File

- **Field List**: The **Field List** displays each of the single fields that are being extracted in a drop-down. Fields can be re-ordered and re-named within the Ordering and Renaming Fields dialog.

- **Add Unique ID to extraction field**: Check to add a unique numerical set of characters to the end of the extracted value. This ensures no two values are identical in the Record Set.
- **Mode**: Determines the origin of the extracted data.
  - **Location**: The contents of the data selection set below will determine the value of the extracted field.
    - **Left**: Defines the start of the data selection to extract.
    - **Right**: Defines the end of the data selection to extract.
    - **Top offset**: The vertical offset from the current pointer location in the Data Sample (**Viewer**).
    - **Height**: The height of the selection box.
    - **Use selection**: Click to use the value (Left, Right and Height) of the current data selection (in the **Viewer**) for the extraction.

      > **Note**
      >
      > If the selection contains multiple lines, only the first line is selected.

    - **Post Function**: Enter a JavaScript expression to be run after the extraction. For example replace("-","") would replace a single dash character inside the extracted string.
      - **Use JavaScript Editor**: Click to display the Script Editor dialog.
    - **Trim**: Select to trim empty characters at the beginning or the end of the field.
    - **Concatenation string**:
    - **Split**: Separate the selection into individual fields based on the **Concatenation string** defined above.

  - **JavaScript** : The result of the JavaScript Expression written below the drop-down will be the value of the extracted field. If the expression contains multiple lines, the last value attribution (variable = "value";) will be the value. See JavaScript API.
- **Type**: The data type of the selected data. Please refer to ...(link to come) for more information.

PDF File

- **Field List**: The **Field List** displays each of the single fields that are being extracted in a drop-down. Fields can be re-ordered and re-named within the Ordering and Renaming Fields dialog.
  - **Add JavaScript Field**: Click to add a new extract field. By default, the field will be a JavaScript -defined field rather than a data extraction, of **String** data type.
  - **Remove Extract Field**: Click to delete the selected extract field from the list.
  - **Order and rename fields**: Click to openOrdering and Renaming Fields.
- **Add Unique ID to extraction field**: Check to add a unique numerical set of characters to the end of the extracted value. This ensures no two values are identical in the Record Set.
- **Mode**: Determines the origin of the data.
  - **Location**: The contents of the data selection set below will be the value of the extracted field. The data selection settings are different depending on the data sample type.
    - **Left**: Defines the start of the data selection to extract.
    - **Right**: Defines the end of the data selection to extract.
    - **Top offset**: The vertical offset from the current pointer location in the Data Sample (**Viewer**).
    - **Height**: The height of the selection box.
    - **Use selection**: Click to use the value (Left, Right and Height) of the current data selection for the extraction.

> **Note**
>
> If the selection contains multiple lines, only the first line is selected.

- **Post Function**: Enter a JavaScript expression to be run after the extraction. For example replace("-","") would replace a single dash character inside the extracted string.
- **Trim**: Select to trim empty characters at the beginning or the end of the field.
- **Type**: The data type of the selected data. Please refer to ...(link to come) for more information.
- **Concatenation string**:
- **Split**: Separate the selection into individual fields based on the **Concatenation string** defined above.
- **Sub-Fields**:

- **JavaScript** : The result of the JavaScript **Expression** written below the drop-down will be the value of the extracted field. If the expression contains multiple lines, the last value attribution (variable = "value";) will be the value. See JavaScript API.
  - **Use JavaScript Editor**: Click to display the Script Editor dialog.
  - **Use selected text**: Inserts the text in the current data selection in the JavaScript Expression. If multiple lines or elements are selected, only the first one is used.
  - **Use selection**: Click to use the value of the current data selection for the extraction.

    > **Note**
    >
    > If the selection contains multiple lines, only the first line is selected.

  - **Type**: The data type of the selected data. Please refer to ...(link to come) for more information.

CSV and Database Files

- **Field List**: The **Field List** displays each of the single fields that are being extracted in a drop-down. Fields can be re-ordered and re-named within the Ordering and Renaming Fields dialog.
  - **Add JavaScript Field**: Click to add a new extract field. By default, the field will be a JavaScript -defined field rather than a data extraction, of **String** data type.
  - **Remove Extract Field**: Click to delete the selected extract field from the list.
  - **Order and rename fields**: Click to openOrdering and Renaming Fields.

- **Add Unique ID to extraction field**: Check to add a unique numerical set of characters to the end of the extracted value. This ensures no two values are identical in the Record Set.
- **Mode**: Determines the origin of the data.
    - **Location**: The contents of the data selection set below will be the value of the extracted field. The data selection settings are different depending on the data sample type.
        - **Column**: Drop-down listing all fields in the Data Sample, of which the value will be used.
        - **Top offset**: The vertical offset from the current pointer location in the Data Sample (**Viewer**).
            - **Use selection**: Click to use the value of the current data selection for the extraction.

                > **Note**
                >
                > If the selection contains multiple lines, only the first line is selected.

        - **Post Function**: Enter a JavaScript expression to be run after the extraction. For example replace("-","") would replace a single dash character inside the extracted string.
            - **Use JavaScript Editor**: Click to display the Script Editor dialog.
        - **Trim**: Select to trim empty characters at the beginning or the end of the field.
        - **Type**: The data type of the selected data. Please refer to ...(link to come) for more information.
    - **JavaScript** : The result of the JavaScript **Expression** written below the drop-down will be the value of the extracted field. If the expression contains multiple lines, the last value attribution (variable = "value";) will be the value. See JavaScript API.
        - **Expression**:
            - **Use JavaScript Editor**: Click to display the Script Editor dialog.
            - **Use selected text**: Inserts the text in the current data selection in the JavaScript Expression. If multiple lines or elements are selected, only the first one is used.

- **Use selection**: Click to use the value of the current data selection for the extraction.

> **Note**
>
> If the selection contains multiple lines, only the first line is selected.

- **Type**: The data type of the selected data. Please refer to ...(link to come) for more information.

XML File

- **Field List**: The **Field List** displays each of the single fields that are being extracted in a drop-down. Fields can be re-ordered and re-named within the Ordering and Renaming Fields dialog.
  - **Add JavaScript Field**: Click to add a new extract field. By default, the field will be a JavaScript -defined field rather than a data extraction, of **String** data type.
  - **Remove Extract Field**: Click to delete the selected extract field from the list.
  - **Order and rename fields**: Click to openOrdering and Renaming Fields.
- **Add Unique ID to extraction field**: Check to add a unique numerical set of characters to the end of the extracted value. This ensures no two values are identical in the Record Set.
- **Based on**: Determines the origin of the data.
  - **Location**: The contents of the data selection set below will be the value of the extracted field. The data selection settings are different depending on the data sample type.
    - **XPath**: The path to the XML field that is extracted.
      - **Use selection**: Click to use the value of the current data selection for the extraction.

> **Note**
>
> If the selection contains multiple lines, only the first line is selected.

  - **Post Function**: Enter a JavaScript expression to be run after the extraction. For example replace("-","") would replace a single dash character inside the extracted string.
    - **Use JavaScript Editor**: Click to display the Script Editor dialog.
  - **Trim**: Select to trim empty characters at the beginning or the end of the field.
  - **Type**: The data type of the selected data. Please refer to ...(link to come) for more information.
- **JavaScript** : The result of the JavaScript **Expression** written below the drop-down will be the value of the extracted field. If the expression contains multiple lines, the last value attribution (variable = "value";) will be the value. See JavaScript API.
  - **Expression**:
    - **Use JavaScript Editor**: Click to display the Script Editor dialog.
    - **Use selected text**: Inserts the text in the current data selection in the JavaScript Expression. If multiple lines or elements are selected, only the first one is used.
    - **Use selection**: Click to use the value of the current data selection for the extraction.

    > **Note**
    >
    > If the selection contains multiple lines, only the first line is selected.

  - **Type**: The data type of the selected data. Please refer to ...(link to come) for more information.

Data Format

The data format defines the precise format for some of the variable types, such as dates and currencies.

- Boolean
- String
- HTML String

- Integer
  - **Negative Sign Before**: Adds a negative (-) sign before negative values in the **Extracted Record**.
  - **Thousand Separator**: Adds the selected character between each thousand position (for example, "1,000,000")
  - **Treat empty as 0:** Considers empty spaces as 0.
- Float
  - **Negative Sign Before**: Adds a negative (-) sign before negative values in the **Extracted Record**.
  - **Decimal Separator**: Determines the character to use for decimal values, generally a dot (.).
  - **Thousand Separator**: Adds the selected character between each thousand position (for example, "1,000,000")
  - **Treat empty as 0:** Considers empty spaces as 0.
- Currency
  - **Negative Sign Before**: Adds a negative (-) sign before negative values in the **Extracted Record**.
  - **Decimal Separator**: Determines the character to use for decimal values, generally a dot (.).
  - **Thousand Separator**: Adds the selected character between each thousand position (for example, "1,000,000")
  - **Currency Sign**: Determines the character to add before the value in the **Extracted Record**.
  - **Treat empty as 0:** Considers empty spaces as 0.
- Date
  - **Date/Time Format**: Determines how the date is read. The format written here needs to correspond with the format in the data selection.
  - **Language**: The language in which the date is written, when relevant. Useful when months or days are written alphabetically.

Note

Default Format Settings can be defined at the user level, at the Datamapper configuration level and/or at the field level. (see Datamapper Default Data Format for more information).

Ordering and Renaming Fields

The **Order and rename fields** dialog appears from the **Extract** step by clicking on the **Rename Fields** button next to the field list. This dialog displays the extracted fields in the currently selected **Extract** step. Field extractions are executed from top to bottom. Using **JavaScript** fields, it is possible to refer to previously extracted fields if they are extracted higher in this list or in previous **Extract** steps in the extraction workflow.

- **Name**: The name of the field. Click the field name and enter a new name to rename the field.
  If you intend to use the field names as metadata in a PReS Workflow process, do not add spaces to field names, as they are not permitted in metadata field names.
- **Value**: Displays the value of the extract field in the current Record.
- **Remove** button : Click to remove the currently selected field.
- **Move Up** button ↑: Click to move the selected field up one position.
- **Move Down** button ↓: Click to move the selected field down one position.

Action Step Properties

The **Action** step can run multiple specific actions one after the other in order. More actions will be available in the future.

Description

This section is collapsed by default in the interface in order to give more screen space to important parts.

Name: The name of the step. This name will be displayed on top of the step's icon in the **Steps** pane.

Comments: The text entered here gives more details on the step and will be displayed in the tooltip appearing when hovering over the step in the **Steps** pane.

Actions

This section lists all actions executed by the step, and their types:

- **Name**: A name by which to refer to the action. This name has no impact on functionality.
- **Type**:
  - **Set property**: Sets the value of a Source Record property which was created in the Preprocessor Step.

    - **Run JavaScript** : Runs a JavaScript expression, giving much more flexibility over the extraction process.

Set Property

Text and PDF Files

- **Property**: Displays a list of Source Record properties set in the Preprocessor Step.
- **Type**: Displays the type of the Source Record property. Read only field.
- **Based on**: Determines the origin of the data.
    - **Location**: The contents of the data selection set below will be the value of the extracted field. The data selection settings are different depending on the data sample type.
        - **Left**: Defines the start of the data selection to extract
        - **Right**: Defines the end of the data selection to extract
        - **Top offset**: The vertical offset from the current pointer location in the Data Sample (**Viewer**).
        - **Height**: The height of the selection box.
        - **Use selection**: Click to use the value of the current data selection for the extraction.

        > **Note**
        >
        > If the selection contains multiple lines, only the first line is selected.

        - **Trim**: Select to trim empty characters at the beginning or the end of the field
    - **JavaScript** : The result of the JavaScript **Expression** written below the drop-down will be the value of the extracted field. If the expression contains multiple lines, the last value attribution (variable = "value";) will be the value. See JavaScript API.
        - **Expression**: The JavaScript expression to run.
        - **Use JavaScript Editor**: Click to display the Edit Script dialog.

        > **Note**
        >
        > Running a JavaScript expression offers many possibilities, for example:

> - Setting Properties and Record Field values using advanced expressions
> - Do complex mathematical operations and calculations
> - More features to come in future versions. For more information, see the JavaScript in DataMapper section.

- **Use selected text**: Inserts the text in the current data selection in the JavaScript Expression. If multiple lines or elements are selected, only the first one is used.
- **Use selection**: Click to use the value of the current data selection for the extraction.

> **Note**
>
> If the selection contains multiple lines, only the first line is selected.

- **Data Format**
  The data format defines the precise format for some of the variable types, such as dates and currencies.
  - Boolean
  - String
  - HTML String
  - Integer
    - **Negative Sign Before**: Adds a negative (-) sign before negative values in the **Extracted Record**.
    - **Thousand Separator**: Adds the selected character between each thousand position (for example, "1,000,000")
    - **Treat empty as 0:** Considers empty spaces as 0.
  - Float
    - **Negative Sign Before**: Adds a negative (-) sign before negative values in the **Extracted Record**.
    - **Decimal Separator**: Determines the character to use for decimal values, generally a dot (.).

    - **Thousand Separator**: Adds the selected character between each thousand position (for example, "1,000,000")
    - **Treat empty as 0:** Considers empty spaces as 0.
- Currency
    - **Negative Sign Before**: Adds a negative (-) sign before negative values in the **Extracted Record**.
    - **Decimal Separator**: Determines the character to use for decimal values, generally a dot (.).
    - **Thousand Separator**: Adds the selected character between each thousand position (for example, "1,000,000")
    - **Currency Sign**: Determines the character to add before the value in the **Extracted Record**.
    - **Treat empty as 0:** Considers empty spaces as 0.
- Date
    - **Date/Time Format**: Determines how the date is read. The format written here needs to correspond with the format in the data selection.
    - **Language**: The language in which the date is written, when relevant. Useful when months or days are written alphabetically.

> **Note**
>
> Default Format Settings can be defined at the user level, at the Datamapper configuration level and/or at the field level. (see Datamapper Default Data Format for more information).

CSV and Database Files

- **Property**: Displays a list of Source Record properties set in the Preprocessor Step.
- **Type**: Displays the type of the Source Record property. Read only field.
- **Based on**: Determines the origin of the data.
    - **Location**: The contents of the data selection set below will be the value of the extracted field. The data selection settings are different depending on the data sample type.
        - **Column**: Drop-down listing all fields in the Data Sample, of which the value will be used.

- **Top offset**: The vertical offset from the current pointer location in the Data Sample (**Viewer**).
- **Use selection**: Click to use the value of the current data selection for the extraction.

> **Note**
>
> If the selection contains multiple lines, only the first line is selected.

- **Trim**: Select to trim empty characters at the beginning or the end of the field

- **JavaScript** : The result of the JavaScript **Expression** written below the drop-down will be the value of the extracted field. If the expression contains multiple lines, the last value attribution (variable = "value";) will be the value. See JavaScript API.
  - **Expression**: The JavaScript expression to run.
  - **Use JavaScript Editor**: Click to display the Edit Script dialog.

> **Note**
>
> Running a JavaScript expression offers many possibilities, for example:
>
> - Setting Properties and Record Field values using advanced expressions
> - Do complex mathematical operations and calculations
> - More features to come in future versions. For more information, see the JavaScript in DataMapper section.

- **Use selected text**: Inserts the text in the current data selection in the JavaScript Expression. If multiple lines or elements are selected, only the first one is used.
- **Use selection**: Click to use the value of the current data selection for the extraction.

> **Note**
>
> If the selection contains multiple lines, only the first line is selected.

- **Data Format**
  The data format defines the precise format for some of the variable types, such as dates and currencies.

  - Boolean
  - String
  - HTML String
  - Integer
    - **Negative Sign Before**: Adds a negative (-) sign before negative values in the **Extracted Record**.
    - **Thousand Separator**: Adds the selected character between each thousand position (for example, "1,000,000")
    - **Treat empty as 0:** Considers empty spaces as 0.
  - Float
    - **Negative Sign Before**: Adds a negative (-) sign before negative values in the **Extracted Record**.
    - **Decimal Separator**: Determines the character to use for decimal values, generally a dot (.).
    - **Thousand Separator**: Adds the selected character between each thousand position (for example, "1,000,000")
    - **Treat empty as 0:** Considers empty spaces as 0.
  - Currency
    - **Negative Sign Before**: Adds a negative (-) sign before negative values in the **Extracted Record**.
    - **Decimal Separator**: Determines the character to use for decimal values, generally a dot (.).
    - **Thousand Separator**: Adds the selected character between each thousand position (for example, "1,000,000")
    - **Currency Sign**: Determines the character to add before the value in the

        Extracted Record.
        - **Treat empty as 0:** Considers empty spaces as 0.
    - Date
        - **Date/Time Format**: Determines how the date is read. The format written here needs to correspond with the format in the data selection.
        - **Language**: The language in which the date is written, when relevant. Useful when months or days are written alphabetically.

> **Note**
>
> Default Format Settings can be defined at the user level, at the Datamapper configuration level and/or at the field level. (see Datamapper Default Data Format for more information).

XML File

- **Property**: Displays a list of Source Record properties set in the Preprocessor Step.
- **Type**: Displays the type of the Source Record property. Read only field.
- **Based on**: Determines the origin of the data.
    - **Location**: The contents of the data selection set below will be the value of the extracted field. The data selection settings are different depending on the data sample type.
        - **XPath**: The path to the XML field that is extracted.
            - **Use selection**: Click to use the value of the current data selection for the extraction.

              > **Note**
              >
              > If the selection contains multiple lines, only the first line is selected.

            - **Trim**: Select to trim empty characters at the beginning or the end of the field

- **JavaScript** : The result of the JavaScript **Expression** written below the drop-down will be the value of the extracted field. If the expression contains multiple lines, the last value attribution (variable = "value";) will be the value. See JavaScript API.
  - **Expression**: The JavaScript expression to run.
  - **Use JavaScript Editor**: Click to display the Edit Script dialog.

    > **Note**
    >
    > Running a JavaScript expression offers many possibilities, for example:
    >
    > - Setting Properties and Record Field values using advanced expressions
    > - Do complex mathematical operations and calculations
    > - More features to come in future versions. For more information, see the JavaScript in DataMapper section.

  - **Use selected text**: Inserts the text in the current data selection in the JavaScript Expression. If multiple lines or elements are selected, only the first one is used.
  - **Use selection**: Click to use the value of the current data selection for the extraction.

    > **Note**
    >
    > If the selection contains multiple lines, only the first line is selected.

- **Data Format**
  The data format defines the precise format for some of the variable types, such as dates and currencies.

  - Boolean
  - String
  - HTML String

- Integer
  - **Negative Sign Before**: Adds a negative (-) sign before negative values in the **Extracted Record**.
  - **Thousand Separator**: Adds the selected character between each thousand position (for example, "1,000,000")
  - **Treat empty as 0:** Considers empty spaces as 0.
- Float
  - **Negative Sign Before**: Adds a negative (-) sign before negative values in the **Extracted Record**.
  - **Decimal Separator**: Determines the character to use for decimal values, generally a dot (.).
  - **Thousand Separator**: Adds the selected character between each thousand position (for example, "1,000,000")
  - **Treat empty as 0:** Considers empty spaces as 0.
- Currency
  - **Negative Sign Before**: Adds a negative (-) sign before negative values in the **Extracted Record**.
  - **Decimal Separator**: Determines the character to use for decimal values, generally a dot (.).
  - **Thousand Separator**: Adds the selected character between each thousand position (for example, "1,000,000")
  - **Currency Sign**: Determines the character to add before the value in the **Extracted Record**.
  - **Treat empty as 0:** Considers empty spaces as 0.
- Date
  - **Date/Time Format**: Determines how the date is read. The format written here needs to correspond with the format in the data selection.
  - **Language**: The language in which the date is written, when relevant. Useful when months or days are written alphabetically.

Note

Default Format Settings can be defined at the user level, at the Datamapper configuration level and/or at the field level. (see Datamapper Default Data Format for more nformation).

Run JavaScript

- **Expression**: The JavaScript expression to run.
- **Use JavaScript Editor**: Click to display the Edit Script dialog.

> **Note**
>
> Running a JavaScript expression offers many possibilities, for example:
>
> - Setting Properties and Record Field values using advanced expressions
> - Do complex mathematical operations and calculations
> - More features to come in future versions. For more information, see the JavaScript in DataMapper section.

- **Use selected text**: Inserts the text in the current data selection in the JavaScript Expression. If multiple lines or elements are selected, only the first one is used.
- **Use selection**: Click to use the value of the current data selection for the extraction.

> **Note**
>
> If the selection contains multiple lines, only the first line is selected.

Repeat Step Properties

The **Repeat** step is a loop that runs more than once. There are multiple types of repeat steps used for different actions. **Repeat** steps do not automatically move the pointer in the file. In order to avoid infinite loops, a **Goto** step must be present within the loop itself.

By default, if an **Extract** step is added within a **Repeat** step, its extraction is made in a Details Table.

Description

This section is collapsed by default in the interface in order to give more screen space to important parts.

Name: The name of the step. This name will be displayed on top of the step's icon in the **Steps** pane.

Comments: The text entered here gives more details on the step and will be displayed in the tooltip appearing when hovering over the step in the **Steps** pane.

Repeat Definition

- **Repeat type**:
  - **While statement is true**: The loop executes while the statement below is true. The statement is evaluated after the loop so the loop will always run at least once.
  - **Until statement is true**: The loop executes until the statement below is true. The statement is evaluated before the loop so the loop may not run at all.
  - **Until no more elements (for Text, CSV, Database and PDF files only)**: The loop executes as long as there are elements left as selected below.
  - **For Each (for XML files only)**: The loop executes for all nodes on a specified level.

  > **Note**
  >
  > When using an XML **For Each** loop, it is not necessary to skip to the repeating node or to have a **Goto**step to jump to each sibling, as this loop takes care of it automatically.

- **Maximum iterations on each line**: Defines the maximum number of iterations occurring at the same position. This expression is evaluated once when entering the loop. The value returned by the expression must be an integer higher than 0.

- **Use JavaScript Editor**: Click to display the Edit Script dialog.

> **Note**
>
> Running a JavaScript expression offers many possibilities, for example:
>
> - Setting **Properties** and **Record Field** values using advanced expressions
> - Do complex mathematical operations and calculations
> - More features to come in future versions. For more information, see the JavaScript in DataMapper section.

Rule Tree

The **Rule tree** section displays the full combination rules (defined below under the **Condition** section) as a tree, which gives an overview of how the conditions work together as well as the result for each of these conditions for the current record or iteration.

Condition

First, the **Condition List** displays the conditions in list form, instead of the tree form above. Three buttons are available next to the list:

- **Add condition**: Click to create a new condition in the list. This will always branch the current condition as an "AND" operator.
- **Delete condition**: Delete the currently selected condition.
- To rename a **Condition**, double click on its name from the **Rule tree** section.

Conditions are made by comparison of two operands using a specific **Operator**.

> **Technical**
>
> Both the **Left** and **Right** operands have the same properties.

Text and PDF Files

- **Based On**:
    - **Position**: The data in the specified position for the comparison.
        - **Left**: The start position for the data selection. Note that conditions are done on the current line, either the current cursor position, or the current line in a **Repeat** step.
        - **Right**: The end position for the data selection.
        - **Top offset**: The vertical offset from the current pointer location in the Data Sample (**Viewer**).
        - **Height**: The height of the selection box.
        - **Use Selection**: Click to use the value of the current data selection for the extraction.
        - **Trim**: Select to trim empty characters at the beginning or the end of the field.
    - **Value**: A specified static text value.
        - **Value**: The text value to use in the comparison.
        - **Use selected text**: Uses the text in the current data selection as the **Value**. If multiple lines or elements are selected, only the first one is used.
    - **Field**: The contents of a specific field in the **Extracted Record**.
        - **Field**: The **Extracted Record** field to use in the comparison.
    - **JavaScript** : The result of a JavaScript **Expression**.
        - **Expression**: The JavaScript line that is evaluated. Note that the last value attribution to a variable is the one used as a result of the expression.
        - **Use JavaScript Editor**: Click to display the Edit Script dialog.
        - **Use selected text**: Inserts the text in the current data selection in the JavaScript **Expression**. If multiple lines or elements are selected, only the first one is used.
    - **Data Property**: The value of a data-level property set in the Preprocessor Step.
    - **Record Property:** One of the local variables that you can create and that are reset for each document as opposed to data variables that are global because they are initialized only once at the beginning of each job.
    - **Automation Property**: The current value of a Document-level property set in the Preprocessor Step.

- **Extractor Property**: The value of an internal extractor variable:
    - **Counter**: The value of the current counter iteration in a **Repeat** step.
    - **Vertical Position**: The current vertical position on the page, either in Measure (PDF) or Line (Text and CSV).
- **Operators**:
  - **is equal to**: The two specified value are identical for the condition to be **True**.
  - **contains**: The first specified value contains the second one for the condition to be **True**.
  - **is less than**: The first specified value is smaller, numerically, than the second value for the condition to be **True**.
  - **is greater than**: The first specified value is larger, numerically, than the second value for the condition to be **True**.
  - **is empty**: The first specified value is empty. With this operator, there is no second value.
  - **Invert condition**: Inverts the result of the condition. For instance, **is empty** becomes **is not empty**.

CSV and Database Files

- **Based On**:
  - **Position**: The data in the specified position for the comparison.
    - **Column**: Drop-down listing all fields in the Data Sample, of which the value will be used.
    - **Top offset**: The vertical offset from the current pointer location in the Data Sample (**Viewer**).
    - **Use Selection**: Click to use the value of the current data selection for the extraction.
    - **Trim**: Select to trim empty characters at the beginning or the end of the field.
  - **Value**: A specified static text value.
    - **Value**: The text value to use in the comparison.
    - **Use selected text**: Uses the text in the current data selection as the **Value**. If multiple lines or elements are selected, only the first one is used.
  - **Field**: The contents of a specific field in the **Extracted Record**.
    - **Field**: The **Extracted Record** field to use in the comparison.

- **JavaScript** : The result of a JavaScript **Expression**.
  - **Expression**: The JavaScript line that is evaluated. Note that the last value attribution to a variable is the one used as a result of the expression.
  - **Use JavaScript Editor**: Click to display the Edit Script dialog.
  - **Use selected text**: Inserts the text in the current data selection in the JavaScript **Expression**. If multiple lines or elements are selected, only the first one is used.
- **Data Property**: The value of a data-level property set in the **Preprocessor** step.
- **Record Property:** One of the local variables that you can create and that are reset for each document as opposed to data variables that are global because they are initialized only once at the beginning of each job.
- **Automation Property**: The current value of a Document-level property set in the **Preprocessor** step.
- **Extractor Property**: The value of an internal extractor variable:
  - **Counter**: The value of the current counter iteration in a **Repeat** step.
  - **Vertical Position**: The current vertical position on the page, either in Measure (PDF) or Line (Text and CSV).

- **Operators**:
  - **is equal to**: The two specified value are identical for the condition to be **True**.
  - **contains**: The first specified value contains the second one for the condition to be **True**.
  - **is less than**: The first specified value is smaller, numerically, than the second value for the condition to be **True**.
  - **is greater than**: The first specified value is larger, numerically, than the second value for the condition to be **True**.
  - **is empty**: The first specified value is empty. With this operator, there is no second value.
  - **Invert condition**: Inverts the result of the condition. For instance, **is empty** becomes **is not empty**.

XML Files

- **Based On**:

  - **Position**: The data in the specified position for the comparison.
    - **XPath**: The path to the XML field that is extracted.
    - **Use Selection**: Click to use the value of the current data selection for the extraction.
    - **Trim**: Select to trim empty characters at the beginning or the end of the field.
  - **Value**: A specified static text value.
    - **Value**: The text value to use in the comparison.
    - **Use selected text**: Uses the text in the current data selection as the **Value**. If multiple lines or elements are selected, only the first one is used.
  - **Field**: The contents of a specific field in the **Extracted Record**.
    - **Field**: The **Extracted Record** field to use in the comparison.
  - **JavaScript** : The result of a JavaScript **Expression**.
    - **Expression**: The JavaScript line that is evaluated. Note that the last value attribution to a variable is the one used as a result of the expression.
    - **Use JavaScript Editor**: Click to display the Edit Script dialog.
    - **Use selected text**: Inserts the text in the current data selection in the JavaScript **Expression**. If multiple lines or elements are selected, only the first one is used.
  - **Data Property**: The value of a data-level property set in the **Preprocessor** step.
  - **Record Property:** One of the local variables that you can create and that are reset for each document as opposed to data variables that are global because they are initialized only once at the beginning of each job.
  - **Automation Property**: The current value of a Document-level property set in the **Preprocessor** step.
  - **Extractor Property**: The value of an internal extractor variable:
    - **Counter**: The value of the current counter iteration in a **Repeat** step.
    - **Vertical Position**: The current vertical position on the page, either in Measure (PDF) or Line (Text and CSV).
- **Operators**:
  - **is equal to**: The two specified value are identical for the condition to be **True**.
  - **contains**: The first specified value contains the second one for the condition to be **True**.

- **is less than**: The first specified value is smaller, numerically, than the second value for the condition to be **True**.
- **is greater than**: The first specified value is larger, numerically, than the second value for the condition to be **True**.
- **is empty**: The first specified value is empty. With this operator, there is no second value.
- **Invert condition**: Inverts the result of the condition. For instance, **is empty** becomes **is not empty**.

Condition Step Properties

Description

This section is collapsed by default in the interface in order to give more screen space to important parts.

Name: The name of the step. This name will be displayed on top of the step's icon in the **Steps** pane.

Comments: The text entered here gives more details on the step and will be displayed in the tooltip appearing when hovering over the step in the **Steps** pane.

Rule Tree

The **Rule tree** section displays the full combination rules (defined below under the **Condition** section) as a tree, which gives an overview of how the conditions work together as well as the result for each of these conditions for the current record or iteration.

Condition

First, the **Condition List** displays the conditions in list form, instead of the tree form above. Three buttons are available next to the list:

- **Add condition**: Click to create a new condition in the list. This will always branch the current condition as an "AND" operator.
- **Delete condition**: Delete the currently selected condition.
- To rename a **Condition**, double click on its name from the **Rule tree** section.

Conditions are made by comparison of two operands using a specific **Operator**.

Technical

Both the **Left** and **Right** operands have the same properties.

Text and PDF Files

- **Based On**:
  - **Position**: The data in the specified position for the comparison.
    - **Left**: The start position for the data selection. Note that conditions are done on the current line, either the current cursor position, or the current line in a **Repeat** step.
    - **Right**: The end position for the data selection.
    - **Top offset**: The vertical offset from the current pointer location in the Data Sample (**Viewer**).
    - **Height**: The height of the selection box.
    - **Use Selection**: Click to use the value of the current data selection for the extraction.
    - **Trim**: Select to trim empty characters at the beginning or the end of the field.
  - **Value**: A specified static text value.
    - **Value**: The text value to use in the comparison.
    - **Use selected text**: Uses the text in the current data selection as the **Value**. If multiple lines or elements are selected, only the first one is used.
  - **Field**: The contents of a specific field in the **Extracted Record**.
    - **Field**: The **Extracted Record** field to use in the comparison.
  - **JavaScript** : The result of a JavaScript **Expression**.
    - **Expression**: The JavaScript line that is evaluated. Note that the last value attribution to a variable is the one used as a result of the expression.
    - **Use JavaScript Editor**: Click to display the Edit Script dialog.
    - **Use selected text**: Inserts the text in the current data selection in the JavaScript **Expression**. If multiple lines or elements are selected, only the first one is used.
  - **Data Property**: The value of a data-level property set in the Preprocessor Step.

    - **Record Property:** One of the local variables that you can create and that are reset for each document as opposed to data variables that are global because they are initialized only once at the beginning of each job.
    - **Automation Property**: The current value of a Document-level property set in the Preprocessor Step.
    - **Extractor Property**: The value of an internal extractor variable:
        - **Counter**: The value of the current counter iteration in a **Repeat** step.
        - **Vertical Position**: The current vertical position on the page, either in Measure (PDF) or Line (Text and CSV).
- **Operators**:
    - **is equal to**: The two specified value are identical for the condition to be **True**.
    - **contains**: The first specified value contains the second one for the condition to be **True**.
    - **is less than**: The first specified value is smaller, numerically, than the second value for the condition to be **True**.
    - **is greater than**: The first specified value is larger, numerically, than the second value for the condition to be **True**.
    - **is empty**: The first specified value is empty. With this operator, there is no second value.
    - **Invert condition**: Inverts the result of the condition. For instance, **is empty** becomes **is not empty**.

CSV and Database Files

- **Based On**:
    - **Position**: The data in the specified position for the comparison.
        - **Column**: Drop-down listing all fields in the Data Sample, of which the value will be used.
        - **Top offset**: The vertical offset from the current pointer location in the Data Sample (**Viewer**).
        - **Use Selection**: Click to use the value of the current data selection for the extraction.
        - **Trim**: Select to trim empty characters at the beginning or the end of the field.

    - **Value**: A specified static text value.
        - **Value**: The text value to use in the comparison.
        - **Use selected text**: Uses the text in the current data selection as the **Value**. If multiple lines or elements are selected, only the first one is used.
    - **Field**: The contents of a specific field in the **Extracted Record**.
        - **Field**: The **Extracted Record** field to use in the comparison.
    - **JavaScript** : The result of a JavaScript **Expression**.
        - **Expression**: The JavaScript line that is evaluated. Note that the last value attribution to a variable is the one used as a result of the expression.
        - **Use JavaScript Editor**: Click to display the Edit Script dialog.
        - **Use selected text**: Inserts the text in the current data selection in the JavaScript **Expression**. If multiple lines or elements are selected, only the first one is used.
    - **Data Property**: The value of a data-level property set in the Preprocessor Step.
    - **Record Property:** One of the local variables that you can create and that are reset for each document as opposed to data variables that are global because they are initialized only once at the beginning of each job.
    - **Automation Property**: The current value of a Document-level property set in the Preprocessor Step.
    - **Extractor Property**: The value of an internal extractor variable:
        - **Counter**: The value of the current counter iteration in a **Repeat** step.
        - **Vertical Position**: The current vertical position on the page, either in Measure (PDF) or Line (Text and CSV).
- **Operators**:
    - **is equal to**: The two specified value are identical for the condition to be **True**.
    - **contains**: The first specified value contains the second one for the condition to be **True**.
    - **is less than**: The first specified value is smaller, numerically, than the second value for the condition to be **True**.
    - **is greater than**: The first specified value is larger, numerically, than the second value for the condition to be **True**.
    - **is empty**: The first specified value is empty. With this operator, there is no second value.

- **Invert condition**: Inverts the result of the condition. For instance, **is empty** becomes **is not empty**.

XML File

- **Based On**:
  - **Position**: The data in the specified position for the comparison.
    - **XPath**: The path to the XML field that is extracted.
    - **Use Selection**: Click to use the value of the current data selection for the extraction.
    - **Trim**: Select to trim empty characters at the beginning or the end of the field.
  - **Value**: A specified static text value.
    - **Value**: The text value to use in the comparison.
    - **Use selected text**: Uses the text in the current data selection as the **Value**. If multiple lines or elements are selected, only the first one is used.
  - **Field**: The contents of a specific field in the **Extracted Record**.
    - **Field**: The **Extracted Record** field to use in the comparison.
  - **JavaScript** : The result of a JavaScript **Expression**.
    - **Expression**: The JavaScript line that is evaluated. Note that the last value attribution to a variable is the one used as a result of the expression.
    - **Use JavaScript Editor**: Click to display the Edit Script dialog.
    - **Use selected text**: Inserts the text in the current data selection in the JavaScript **Expression**. If multiple lines or elements are selected, only the first one is used.
  - **Data Property**: The value of a data-level property set in the Preprocessor Step.
  - **Record Property:** One of the local variables that you can create and that are reset for each document as opposed to data variables that are global because they are initialized only once at the beginning of each job.
  - **Automation Property**: The current value of a Document-level property set in the Preprocessor Step.
  - **Extractor Property**: The value of an internal extractor variable:
    - **Counter**: The value of the current counter iteration in a **Repeat** step.
    - **Vertical Position**: The current vertical position on the page, either in Measure (PDF) or Line (Text and CSV).

- **Operators**:
  - **is equal to**: The two specified value are identical for the condition to be **True**.
  - **contains**: The first specified value contains the second one for the condition to be **True**.
  - **is less than**: The first specified value is smaller, numerically, than the second value for the condition to be **True**.
  - **is greater than**: The first specified value is larger, numerically, than the second value for the condition to be **True**.
  - **is empty**: The first specified value is empty. With this operator, there is no second value.
  - **Invert condition**: Inverts the result of the condition. For instance, **is empty** becomes **is not empty**.

Goto Step Properties

The **Goto** step moves the pointer within the data sample to a specific position or one that is relative to the current position.

Description

This section is collapsed by default in the interface in order to give more screen space to important parts.

Name: The name of the step. This name will be displayed on top of the step's icon in the **Steps** pane.

Comments: The text entered here gives more details on the step and will be displayed in the tooltip appearing when hovering over the step in the **Steps** pane.

Goto Definition

Text File

- **Target Type**: Defines the type of jump .
  - **Line**: Jumps a certain number of lines or to a specific line.
    - **From**: Defines where the jump begins:
      - **Current Position**: The **Goto** begins at the current cursor position.
      - **Top of record**: The **Goto**begins at line 1 of the source record.

    - **Move by**: Enter the number of lines or pages to jump.
- **Page**: Jumps between pages or to a specific page.
    - **From**: Defines where the jump begins:
        - **Current Position**: The **Goto**begins at the current cursor position.
        - **Top of record**: The **Goto**begins at line 1 of the source record.
    - **Move by**: Enter the number of lines or pages to jump.
- **Next line with content**: Jumps to the next line that has contents, either anywhere on the line or in specific columns.
    - **Inspect entire page width**: When checked, the **Next line with content** and **Next occurrence of** options will look anywhere on the line. If unchecked, options appear below to specify in which area of each line the **Goto**step checks in:
        - **Left**: The starting column, inclusively.
        - **Right**: The end column, inclusively.
        - **Use selection**: Click while a selection is made in the Data Viewer to automatically set the left and right values to the left and right edges of the selection.
- **Next occurrence of**: Jumps to the next occurrence of specific text or a text pattern, either anywhere on the line or in specific columns.
    - **Inspect entire page width**: When checked, the **Next line with content** and **Next occurrence of** options will look anywhere on the line. If unchecked, options appear below to specify in which area of each line the **Goto**step checks in:
        - **Left**: The starting column, inclusively.
        - **Right**: The end column, inclusively.
        - **Use selection**: Click while a selection is made in the Data Viewer to automatically set the left and right values to the left and right edges of the selection.
    - **Expression**: Enter the text or **Regex** expression to look for on the page.
    - **Use selection**: Click while a selection is made in the Data Viewer to copy the contents of the first line of the selection into the **Expression** box.

        - **Use regular expression**: Check so that the **Expression** box is treated as a regular expression instead of static text. For more information on using **Regular Expressions (regex)**, see the Regular-Expressions.info Tutorial.

PDF File

- **Target Type**: Defines the type of jump .
    - **Physical distance**:
        - **From**: Defines where the jump begins:
            - **Current Position**: The **Goto** begins at the current cursor position.
            - **Top of record**: The **Goto**begins at line 1 of the source record.
        - **Move by**: Enter distance to jump.
    - **Page**: Jumps between pages or to a specific page.
        - **From**: Defines where the jump begins:
            - **Current Position**: The **Goto**begins at the current cursor position.
            - **Top of record**: The **Goto**begins at line 1 of the source record.
        - **Move by**: Enter the number pages to jump.
    - **Next line with content**: Jumps to the next line that has contents, either anywhere on the line or in specific columns.
        - **Inspect entire page width**: When checked, the **Next line with content** and **Next occurrence of** options will look anywhere on the line. If unchecked, options appear below to specify in which area of each line the **Goto**step checks in:
            - **Left**: The starting column, inclusively.
            - **Right**: The end column, inclusively.
            - **Use selection**: Click while a selection is made in the Data Viewer to automatically set the left and right values to the left and right edges of the selection.
    - **Next occurrence of**: Jumps to the next occurrence of specific text or a text pattern, either anywhere on the line or in specific columns.
        - **Inspect entire page width**: When checked, the **Next line with content** and **Next occurrence of** options will look anywhere on the line. If unchecked, options appear below to specify in which area of each line the **Goto**step

checks in:

    - **Left**: The starting column, inclusively.
    - **Right**: The end column, inclusively.
    - **Use selection**: Click while a selection is made in the Data Viewer to automatically set the left and right values to the left and right edges of the selection.

- **Expression**: Enter the text or **Regex** expression to look for on the page.
- **Use selection**: Click while a selection is made in the Data Viewer to copy the contents of the first line of the selection into the **Expression** box.
- **Use regular expression**: Check so that the **Expression** box is treated as a regular expression instead of static text. For more information on using **Regular Expressions (regex)**, see the Regular-Expressions.info Tutorial.

CSV File

- **From (CSV files)**: Defines where the jump begins:
  - **Current Position**: The **Goto** begins at the current cursor position.
    - **Move by**: Enter the number of lines or pages to jump.
  - **Top of record**: The **Goto**begins at line 1 of the source record.
    - **Move to**: Enter the number of lines or pages to jump.

XML File

- **Destination (XML files)**: Defines what type of jump to make:
  - **Sibling element**: Jumps the number of siblings (nodes at the same level) defined in the **Move by**option.
  - **Sibling element with same name**: Jumps the number of same name siblings (nodes at the same level of which the node is the same name) defined in the **Move by**option.
  - **Element, from top of record**: Jumps to the specified node. The XPATH in the **Absolute XPATH**option starts from the root node defined by **/**.
  - **Element from current position**: Jumps to a position relative to the current position of the cursor. The XPATH in the **Relative XPATH**option defines where to go,**../**goes up a level,**./**refers to the current level.

    - **Level Up/Down**: Jumps up or down one node level (up to the parent, down to a child). The number of levels to change is defined in the **Move by**option.

Postprocessor Properties

The Postprocessor step does not run for every Source Record in the Data Sample. It runs once, at the end of the Steps, after all records have been processed.

Postprocessor

The Postprocessor section defines what postprocessors run on the Data Sample at the end of the data mapping workflow. Each Postprocessor runs in turn, using the result of the previous one as an input.

- **Name**: The name to identify the Postprocessor.
- **Type**: The type of Postprocessor. Currently there is a single type available.
    - **JavaScript** : Runs a JavaScript Expression to modify the Data Sample. See JavaScript API.
        - **Use JavaScript Editor**: Click to display the Edit Script dialog.
- **Add Postprocessor**: Click to add a new Postprocessor. Its settings can be modified once it is added.
- **Remove Postprocessor**: Click to remove the currently selected Postprocessor.
- **Move Up**: Click to move the Postprocessor up one position.
- **Move Down**: Click to move the Postprocessor down one position.
- **Export**: Click to export the current Postprocessor configuration and content to a file.
- **Import**: Click to import a Postprocessor configuration and content from an external file.

Postprocessor definition

JavaScript

- **Expression**: The JavaScript expression that will run on the Data Sample. See JavaScript API.
- **Use JavaScript Editor**: Click to display the Script Editor dialog.
- **Use selected text**: Uses the text in the current data selection as the Value. If multiple lines or elements are selected, only the first one is used.

Description

This section is collapsed by default in the interface in order to give more screen space to important parts.

Name: The name of the step. This name will be displayed on top of the step's icon in the **Steps** pane.

Comments: The text entered here gives more details on the step and will be displayed in the tooltip appearing when hovering over the step in the **Steps** pane.

Data Types

Each piece of data in PReS Connect, once it has been extracted from the Source Record, will always be handled using Data Types. Data types determine how the data is handled when reading and writing values in properties and extracted record fields, as well as when comparing values. For example, if a Date field is compared with a Date source record property, the comparison will be done on dates (latest date being a "higher" value).

Here are the data types available in all modules of PReS Connect.

- Boolean
- String
- HTMLString
- Integer
- Float
- Currency
- Date
- Object

Note

The Object data type is only available in the DataMapper module. It is created in the Properties section of the Preprocessor step, and can be used throughout the data mapping configuration.

Boolean Data Type

Booleans are a simple True/False data type often used in conditions and comparisons.

Defining Boolean Values

- **Pre-Processor**: Specify the "Type" as "Boolean" and set a default value of either **true;** or **false;**
- **Extraction**: Specify the "Type" as "Boolean". The field value must be**true**or**false**.
- **JavaScript Expression**: Set the desired value to either **true;** or **false;**
  Example: **record.fields["isCanadian"] = true;**

> **Note**
>
> The value must be *true* , all in lowercase. Any variation in case (True, TRUE) will not work.

Boolean Expressions

Boolean values can also be set using an expression of which the result is true or false. This is done using operators and comparisons.

Example: **record.fields["isCanadian"] = (extract("Country") == "CA");**

For more information on JavaScript *comparison* and *logical operators*, please see w3schools.com or developer.mozilla.org.

String Data Type

Strings contain textual data. Strings do not have any specific meaning, which is to say that their contents are never interpreted in any way.

Defining String Values

- **Pre-Processor**: Specify the "Type" as "String" and set a default value as any text between quotes, such as **"This is my text";**
- **Extraction**: Specify the "Type" as "String". The field value will be extracted and treated as a string.
- **JavaScript Expression**: Set the desired value to any string between quotes.
  Example: **record.fields["countryOfOrigin"] = "Canada";**

Building String Values

Strings values can be made up of more than just a series of characters between quotes. Here are a few tips and tricks to build strings:

- Both single and double quotes can be used to surround strings and they will act in precisely the same manner. So, **"this is a string"** and **'this is a string'** mean the same thing. However, it's useful to have both in order to remove the need for escaping characters. For instance, **"I'm fine!"** works, but **'I'm fine!'** does not since only **'I'** is properly interpreted. **'I\'m fine!'** works (escaping the ' with a \).
- It is possible to put more than one string, as well as variables containing strings, by concatenating them with the **+**operator. For example,**"Hello " + sourceRecord.property.FirstName + ", nice to meet you!"**.
- Adding more data to an existing string variable or field is possible using a combination of concatenation and assignment. For example, if **var myVar = "Is this the real life";**, and**myVar += " or is this just fantasy?";**, the value of**myVar**will be, obviously,**"Is this the real life or is this just fantasy?"**.

For more information on string variables, see quicksmode.org.

HTML String Data Type

HTML Strings contain textual data that includes HTML Markup. They are essentially the same as String values except in cases where HTML can be treated differently. For instance, when using a script to place an HTML String field into a template, the contents will be HTML instead of plain text.

Example: Assuming a value of "He said <b>WOW!</b>"; , if the data type is String and placed on the page, it will display exactly as "He said <b>WOW!</b>" (without the quotes). If the data type is HTMLString, it will display as "He said **WOW!**" (again, without the quotes).

Considering this is the only difference, for more information on how to create and use HTML String values, see String values.

Integer Data Type

Integers are signed, numeric, whole 64bit numbers whose value ranges from -(2^63) to (2^63). Integers are the numerals with the highest precision (and the fastest processing speed) of all since they are never rounded.

Defining Integer Values

- **Pre-Processor**: Specify the "Type" as "Integer" and set a default value as a number, such as **42;**
- **Extraction**: Specify the "Type" as "Integer". The field value will be extracted and treated as an integer.
- **JavaScript Expression**: Set the desired value to any integer value.
  Example: **record.fields["AnswerToEverything"] = 42;**

Building Integer Values

Integers can be set through a few methods, all of which result into an actual integer result.

- **Direct attribution**: Assign an integer value directly, such as **42**,**99593463712**or**data.extract("TotalOrdered")**.
- **Mathematical operations**: Assign the result of any mathematical operation. For example:**22 + 51**,**3*6**,**10/5**or**sourceRecord.property.SubTotal**. For more information on mathematics in JavaScript , see w3Schools - Mathematical Operators. For more advanced mathematical functions, see w3schools - Math Object.

> **Note**
>
> When adding numbers that are not integers, for instance **4.5 + 1.2** , a round towards zero rounding is applied after the operation was made. In the previous example, the result, **5.7**, is rounded to **5**. In another example, **-1.5 - 1** results in **-2**

Float Data Type

Floats are signed, numeric, floating-point numbers whose value has 15-16 significant digits. Routinely used for calculations. Note that floats are inherently imprecise: their accuracy varies according to the number of significant digits being requested.

Defining Float Values

- **Pre-Processor**: Specify the "Type" as "Float" and set a default value as a number with decimal points, such as **546513.8798463;**
- **Extraction**: Specify the "Type" as "Float". The field value will be extracted and treated as a float.
- **JavaScript Expression**: Set the desired value to any float value.
  Example: **record.fields["PreciseTaxSubtotal"] = 27.13465;**

Building Float Values

Float values can be the result of direct attribution or mathematical operations just like Integer values.

Currency Data Types
A signed, numeric, fixed-point 64-bit number with 4 decimals. Values range from *-922 337 203 685 477.5808* to *922 337 203 685 477.5808*. This data type is routinely used for financial calculations: it is as precise as integers.

Defining Currency Values

- **Pre-Processor**: Specify the "Type" as "Currency" and set a default value as a number with up to 4 decimal points, such as **546513.8798**
- **Extraction**: Specify the "Type" as "Currency" and configure how the value will be read from the Source Record (see Extract Step). The field value will be extracted and treated as a float.
- **JavaScript Expression**: Set the desired value to any float value.
  Example: **record.fields["PreciseTaxSubtotal"] = 27.13465;**

While currency values can be set to up to 4 significant digits, only 2 are displayed on screen.

Building Currency Values

Currency values can be the result of direct attribution or mathematical operations just like Integer and Float values.

Date Data Type

Dates are values that represent a specific point in time, precise up to the second. They can also be referred to as datetime values. While dates are shown visually under the system's regional settings, in reality they are stored unformatted.

> **Note**
>
> The **Date** variable is stored in **Connect** database with zero time zone offset, which makes it possible to convert the time correctly in any location. PReS Worfklow, however, shows the date/time as it is stored database (with 0 time zone offset). This is expected behavior for the moment and the zone offset must be calculated manually in PReS Workflow.

Defining Date Values

Because dates can be in many different formats and variations, the use of the JavaScript Date() object is necessary when creating dates through a JavaScript expression. For more information, see w3schools - Date Object.

- **Pre-Processor**: Specify the "Type" as "Date" and set a default value as a date value in any desired format, such as **"[TBD]";**
- **Extraction**: Specify the "Type" as "Date" and set the **Data Format**, especially the "DateTime Format" that defines how the date is read. The field value will be extracted and treated as a **Date**.
- **JavaScript Expression**: Set the desired value to any date value.
  Example: **record.fields["InvoiceDate"] = "[TBD]";**

Object Data Type

Objects holds addresses that refer to objects. You can assign any reference type (string, array, class, or interface) to an Object variable. An Object variable can also refer to data of any value type (numeric, Boolean, Char, Date, structure, or enumeration).

Defining Object Values

- **Pre-Processor**: Specify the "Type" as "Object" and set a default value as a semi-colon.

Toolbar

In the DataMapper module, the following buttons are available in the top toolbar:

File Manipulation Buttons

- **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 data mapping configuration.
- **Save** : Saves the current data mapping configuration. If the configuration has never been saved, the **Save As...** dialog is displayed.

Step Manipulation

Note

All steps except JavaScript require an active data selection in the *Data Viewer*.

- **Add Extract Step** : Adds an Extract Step with one or more extract fields. If more than one line or field is selected in the Data Viewer, each line or field will have an extract field.
- **Add Goto Step** : Adds a Goto step that moves the selection pointer to the beginning of the data selection. For instance if an XML node is selected, the pointer moves to where that node is located.
- **Add Condition Step** : Adds a condition based on the current data selection. The "True" branch gets run when the text is found on the page. Other conditions are available in the step properties once it has been added.
- **Add Repeat Step** : Adds a loop that is based on the current data selection, and depending on the type of data. XML data will loop on the currently selected node, CSV loops for all rows in the record. In Text and PDF data, if the data selection is on the same line as the cursor position, the loop will be for each line until the end of the record. If the

data selection is on a lower line, the loop will be for each line until the text in the data selection is found at the specified position on the line (e.g. until "TOTAL" is found).

- **Add Extract Field**: Adds the data selection to the selected **Extract** step, if an extract step is currently selected. If multiple lines, nodes or fields are selected, multiple extract fields are added simultaneously.
- **Add Action Step**: Adds a step to create a custom JavaScript snippet. See the JavaScript API for more details.
- **Cut Step**: Removes the currently selected step and places it in the clipboard. If the step is a Repeat or a Condition, all steps under it are also placed in the clipboard. If there is already a step in the clipboard, it will be overwritten.
- **Copy Step**: Places a copy of the currently selected step in the clipboard. The same details as the Cut step applies.
- **Paste Step**: Takes the step or steps in the clipboard and places them after the currently selected step.
- **Delete Step**: Deletes the currently selected step. If the step is a Repeat or Condition, all steps under it are also deleted.
- **Ignore Step**: Click to set the step to be ignored (aka disabled). Disabled steps do not run when in DataMapper and do not execute when the data mapping configuration is executed in Workflow. However, they can still be modified normally.
- **Validate All Records**: Runs the process on all records and verifies that no errors are present in any of the records. Errors are displayed in the Messages Pane.
- **Add Data Sample**: Displays a dialog to open a new Data Source to add it as a Data Sample in the data mapping configuration. Data Samples are visible in the Settings Pane.

Shortcut Keys

You can add a step by pressing a specific key on the keyboard:

Extract: F6

Goto: F7

Conditional: F8

Repeat: F9

Action: F11

Welcome Screen

The **Welcome Screen** appears when first starting up PReS Connect. It offers some useful shortcuts to resources and to recent documents and data mapping configurations.

The Welcome Screen can be brought back in two ways:

- The **Welcome Screen** button in the Toolbars.
- From the Menus in **Help**, **Welcome Screen**.

Contents

- **Activation**: Click to open the **Objectif Lune Web Activation Manager**.
- **Release Notes**: Opens the current **Release Notes** for PReS Connect.
- **Website**: Opens the PReS Connect website.
- **Take A Tour**: Click to open the YouTube Playlist giving you a tour of the software.
- **Use the DataMapper to...**:
  - **Create a New Configuration**: Opens the Creating a New Configuration screen.
  - **Open an Existing Configuration**: Click to open the standard **Browse** dialog to open an existing data mapping configuration.
  - **Recent Configurations**: Lists recently used configurations. Click any configuration to open it in the DataMapper module.
- **Use the Designer to...**:
  - **Create a New Template**: Opens the Creating a new Template wizard.
  - **Browse Template Wizards**: Displays a list of available template wizards, producing pre-made templates with existing demo content. These are the same that are found at the bottom of the Creating a new Template wizard.
  - **Open an Existing Template**: Click to open the standard **Browse** dialog to open an existing template.
  - **Recent Templates**: Lists recently used templates. Click any template to open it in the Designer module.
- **Other Resources**:
  - **Documentation**: Opens this documentation.
  - **Courses (OL Learn)**: Opens the Objectif Lune e-Learning Center.
  - **User Forums**: Opens the Questions & Answer forums.

The Designer

The Designer is a WYSIWIG (what you is what you get) editor that lets you create templates for various output channels: Print, Email and Web. 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 extracted via the DataMapper.More advanced users may use native HTML, CSS and JavaScript.

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

- "Basic Steps" on page 215. These are the basic steps for creating and developing a template.
- "Features" on page 217. These are some of the key features in the Designer.
- "Designer User Interface" on page 449. This section 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.

...

API

Designer Scripts API

This page describes the different features available in scripts created inside the Scripts pane. See "Write your own scripts" on page 373.

Features that are only available in Control Scripts are listed in the Control Script API. See "Control Script API" on page 208.

Objects

results	The HTML element or set of HTML elements that match the selector specified in the script editor.
record	The current record in the main data set.
logger	Global object that allows you to log messages.
locale	Defines which locale to use.
"formatter" on page 197	Global object that allows you to format values.
automation	Automation properties
merge	Merge properties

Global functions

Skin/Formats/CrossReferencePrintFormat("loadhtml() Global function that replaces the content (inner html) of each matched element in the result set, alternatively load the data into a variable. The location should be an URL or a relative file path. loadhtml(location)loadhtml(location, selector) Loadhtml() is cached per batch run (based on the URL) in print/email. loadhtml(location) Loads all HTML from the 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 a HTML file residing in the Snippets folder on the Resources panel.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 variableThe 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	Loads HTML data from a HTML (snippet). The returned HTML can be placed into a variable or into a set of HTML elements.

before inserting the content into the page. The second find command also adds formatting to the replacing 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); This last script loads a snippet into a variable and retrieves an element from the snippet using query().var mysnippet = loadhtml('snippets/text-root-wrapped.html'); var subject = query("#subject", mysnippet).text(); results.append("<p style='font-weight: bold;'>" + subject + "</p>");loadhtml(location, selector) Retrieve specific content from the filename. location String; the location can be absolute or relative to the section/context. Use: snippets/<snippet-name> to retrieve the content from a HTML file residing in snippets folder of the Resources panel.selectorString. The supplied selector should conform to CSS selector syntax and allows you to retrieve only the content of matching elements.ExamplesThis script loads a specific element from the snippet.var mysnippet = loadhtml('snippets/snippet-selectors.html','#item3'); results.replaceWith(mysnippet);This script loads the children of the selected element.var snippet = loadhtml('snippets/snippet.html','foobar').children(); results.replaceWith(snippet);Another example is given in the following how-to: Using a selector to load part of a snippet." on page 1)	
"loadjson()" on page 200	Loads json data from an url. This is an simple way to retrieve content from external systems.
"query()" on page 206	Performs a query in the

	template's contents and creates a new result set containing the HTML elements that match the given CSS selector.

Functions

All these functions can be used with an HTML element or a set of HTML, such as the `results` (the HTML element or set of HTML elements that match the selector specified in the script editor) or the result set returned by a query (see "query()" on page 206).

"add()" on page 176	Adds elements to a set of HTML elements.
"addClass()" on page 177	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 178	Inserts content after each element in a set of HTML elements..
"append()" on page 180	Inserts content at the end of each element in a set of HTML elements.
"attr()" on page 183	Change the given attribute of the element or set of HTML elements with the given value.
"before()" on page 184	Inserts content before an element or before each element in a set of HTML elements.

"css()" on page 188	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.
"children()" on page 186	Returns the immediate children of an HTML element.
"clone()" on page 186	Returns a new result set containing a copy of each element in a set of HTML elements.
filter()	Returns a subset of the current result set.
find()	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.
hasClass()	Returns `true` if the first element in this result set has the specified class.
hide()	Hides the HTML element or set of HTML elements.
html()	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.
is(selector)	Returns true if at least one of the elements in a set of HTML elements matches the supplied CSS selector.
parent()	Returns the parents of the elements in a set of HTML elements.
"prepend()" on page 203	Inserts content at the beginning of an HTML element or of each element in a set of HTML elements.
"CopyFile()" on page 207	Removes an HTML element or a set of HTML elements from the document.
removeAttr()	Removes the specified attribute from each element in this result set.
"removeClass	Removes the specified class from an element or from each element in a

()" on page 207	set of HTML elements. Has no effect if the class is not present.
"replaceWith ()" on page 208	Replaces an HTML element or a set of HTML elements (with a snippet, for example). Returns the result set.
show()	Shows the HTML element or a set of HTML elements.
text()	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.

Examples of iterator functions

"Each" on page 194	A generic iterator function, to iterate over the elements in the result set
"For...in" on page 196	Iterates over the enumerable properties of an object, in arbitrary order. For each distinct property, statements can be executed.

add()

The add() function allows you to add elements to a set of HTML elements that match the selector of the script or of another query in the template (see "query()" on page 206).

add(content)

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	<p id="chapters">{{chapters}}</p>	<h1>Chapter 1</h1> <p>Lorem ipsum...</p> <h1>Chapter 2</h1> <p>Lorem ipsum...</p> <h1>Chapter 3</h1> <p>Lorem ipsum...</p> <h1>Chapter 4</h1> <p>Lorem ipsum...</p>

addClass()

Adds the specified class(es) to each element in a set of HTML elements that match the selector of the script or of another query in the template (see "query()" on page 206). This has no effect if the class is already present.

addClass(classname)

Adds the specified class(es) to each element in a result set. 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	<p>Hello world</p>	<p class="foo bar">Hello world</p>

The following script adds two class names to a paragraph.

```
results.addClass("foo bar");
```

Selector	Matched element	Matched element after script execution
p	<p>Hello world</p>	<p class="foo bar">Hello world</p>

after()

Insert content after 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 206). See also: "before()" on page 184.

after(content)

Insert content after 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 206). After 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 <span> 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
<p id="salesrep">Peter Parker</p>	<p id="salesrep">Peter Parker</p> <p>Lorem ipsum</p>

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
<p id="salesrep">Peter Parker</p>	<p id="salesrep" style="color: red;">Peter Parker</p> <p>Lorem ipsum</p>

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
<p id="salesrep">Peter Parker</p>	<p id="salesrep">Peter Parker</p> <p>Lorem ipsum</p>

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

<p>Lorem dolor sit amet, consectetur adipiscing elit.</p>	<p>Lorem <span>ipsum</span> dolor sit amet, consectetur adipiscing elit.</p>

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**
<p id="salesrep">Peter Parker</p>	<p id="salesrep">Peter Parker</p> <span>Lorem Ipsum</span>

append()

Insert content at the end of each element in the set of each element in a set of HTML elements that match the selector of the script or of another query in the template (see "query()" on page 206). See also: "prepend()" on page 203.

append(content)

Insert content as the last 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 206). `Append` 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 <span> 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	<div id="box"> <h1>Personal information</h1> </div>	<div id="box"> <h1>Personal information</h1> <p>Peter Parker</p> </div>

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	<div> <h1>Personal information</h1> <p class="name"><b>Name: </b></p> </div>	<div> <h1>Personal information</h1> <p class="name"><b>Name: </b><span>Peter Parker</span></p> </div>

This script's selector is <div>, 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	<div> <h1>Personal information</h1> </div> <div> <h1>Personal information</h1> </div>	<div> <h1>Personal information</h1> <p>Peter Parker</p> </div> <div> <h1>Personal information</h1> <p>Peter Parker</p> </div>

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	<div id="box"> <h1>Personal information</h1> </div>	<div id="box"> <h1>Personal information</h1> <p>Peter Parker</p> </div>

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**
<div id="box"> <h1>Personal information</h1> </div>	<div id="box"> <h1>Personal information</h1> <p>Peter Parker</p> </div>

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**
<div id="box"> <h1>Personal information</h1> </div>	<div id="box" style="color: red;"> <h1>Personal information</h1> <p>Peter Parker</p> </div>

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()

Returns the value of the specified attribute of the first element in a result set, or sets the value of the specified attribute of each element in a result set.

- attr(attributeName)
- attr(attributeName, value)

attr(attributeName)

Returns the value of the specified attribute of the first element in a result set.

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 element with the ID #table1 and stores its background color in a variable.

```
var backgroundcolor = query("#table1").attr("bgcolor");
```

attr(attributeName, value)

Sets the value of the specified attribute 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 a table cell in an element with the ID #calloutbox and sets its text color to red.

```
query("#callout td").attr('bgcolor' , 'red');
```

The following script sets the background color of a specific table cell 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()

Insert content before 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 206). See also: "after()" on page 178.

before(content)

Before(content) inserts content before each element in the set of elements that match the script's selector. Before() 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 <span> 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	<p id="salesrep">Peter Parker</p>	<p>Lorem ipsum</p> <p id="salesrep">Peter Parker</p>

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
<p id="salesrep">Peter Parker</p>	<p>Lorem ipsum</p> <p id="salesrep">Peter Parker</p>

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
<p id="salesrep">Peter Parker</p>	<p >Lorem ipsum</p> <p id="salesrep" style="color: red;">Peter Parker</p>

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
<p>ipsum dolor sit amet, consectetur adipiscing elit.</p>	<p><span>Lorem </span>ipsum dolor sit amet, consectetur adipiscing elit.</p>

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
<p>ipsum dolor sit amet, consectetur adipiscing elit.</p>	<span>Lorem Ipsum</span> <p id="salesrep">Peter Parker</p>

children()

Returns the immediate children (inner HTML) of the elements in a result set.

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()

Returns a new result set containing a deep copy of each element in a result set. To duplicate an existing DOM element, clone it before calling append(); see "append()" on page 180.

Examples

This script performs an iteration over the elements in the results

```
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.

```
for(var i = 0; i < record.tables['detail'].length; i++) {

var row = results.clone(); //Clone our boilerplate row

row.find('@ItemNumber@').text( record.tables['detail'][i].fields
["ItemNumber"]);
row.find('@ItemOrdered@').text( record.tables['detail'][i].fields
["ItemOrdered"]);
row.find('@ItemTotal@').text( record.tables['detail'][i].fields
["ItemTotal"]);
row.find('@ItemDesc@').text( record.tables['detail'][i].fields
["ItemDesc"]);
row.find('@nr@').text( i );

results.parent().append(row);
}

// Hide our boilerplate row (note that this doesn't really delete
the row).
results.hide();
```

css()

Get the value of a style property for the first 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 206), or set one or more CSS properties for every element in the set.

- css(propertyName)
- css(propertyName, value)
- css(properties

css(propertyName)

Returns the value of the specified CSS property.

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(propertyName, value)

Function to set a CSS property.

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.

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'});
```

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 418.

> **Tip**
>
> To format a date from a `date` field in the record set, you can enter a formatting pattern directly in the Text Script Wizard; see "Using the Text Script Wizard" on page 335, "Formatting variable data" on page 338 and Date and time patterns).

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. 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. 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. 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

The following script passes the value of a field in the record set to the date() function. This will only work if the type of the field has been set to Date in the Data Mapping Configuration and if the field contains a valid date.

```
var myDate = formatter.date(records.fields.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.

Creating a Date object from a string

In a Data Mapping Configuration you can set the type of a field to Date, but when you open a data file or database in the Designer without a Data Mapping Configuration, 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 http://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");
```

Each

A generic iterator function, to iterate over the elements in the result set.

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 two scripts demonstrate a simple iteration over the elements in the `results` (the set of HTML elements that match the selector of the script).

This 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 188.)

```
results.each(function(index){
        results[index].css('background-color','red');
});
```

The following 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	<p></p> <p></p> <p></p>	<p>3</p> <p>1</p> <p>7</p>

This 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	<p></p> <p></p> <p></p>	<p>0</p> <p>1</p> <p>2</p>

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 http://www.w3schools.com/tags/att_global_data.asp.

Selector	Matched element	Matched element after script execution
p	<p data-translate="first"></p> <p data-translate="last"></p> <p data-translate="email"></p>	<p>primero</p> <p>último</p> <p>dirección de correo electrónico</p>

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> <p>first</p> <p>last</p> <p>email</p>

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> <p>Peter</p> <p>Parker</p> <p>pparker@localhost.com</p>

This script iterates over rows in a detail table and adds the contents of the 'country' field to a paragraph.

```
for(var i in record.tables['countries']) {
        results.after("<p>" + record.tables['countries'][i].fields
['country'] + "</p>");
}
```

Selector	Matched element	Matched element after script execution
#countries	<h1 id="countries">Countries</h1>	<h1 id="countries">Countries</h1> <p>The Netherlands</p> <p>Canada</p> <p>Australia</p>

This script iterates over rows in a detail table and adds the contents of the 'ItemID2' field to an option. The <option> tag defines an option in a select list in an HTML form.

```
for(var i in record.tables['detail']) {
        var str = record.tables['detail'][i].fields["ItemID2"];
        results.append("<option value='" + str + "'>" + str +
"</option>");
}
```

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 335 and "Formatting variable data" on page 338.

> **Note**
>
> The TextFormatter object is now deprecated and will eventually be removed.

Functions

• currency() • currencyNoSymbol() • grouped()	The currency, grouped, integer and number functions allow you to format a number, possibly with a custom pattern. See

• integer() • integerUngrouped() • number() • numberUngrouped()	"Number functions" on page 202.
• date() • dateLong() • dateMedium() • dateShort() • dateTime() • dateTimeLong() • dateTimeMedium() • dateTimeShort() • timeLong() • timeMedium() • timeShort()	The date, dateTime and time functions allow you to format a date and/or time. See "Date, date/time and time functions" on page 189.
• lowerCase() • upperCase() • properCase()	The text formatting functions transform all characters to lowercase (`lowerCase`) or uppercase (`upperCase`) or transforms the first character of each word to uppercase and all other characters to lowercase (`properCase`).

loadhtml()

Global function that replaces the content (inner html) of each matched element in the result set, alternatively load the data into a variable. The location should be an URL or a relative file path.

- loadhtml(location)
- loadhtml(location, selector)

> **Note**
>
> Loadhtml() is cached per batch run (based on the URL) in print/email.

loadhtml(location)

Loads all HTML from the 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 a HTML file residing in the Snippets folder on the Resources panel.

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('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);
```

This last script loads a snippet into a variable and retrieves an element from the snippet using query().

```
var mysnippet = loadhtml('snippets/text-root-wrapped.html');
var subject = query("#subject", mysnippet).text();
results.append("<p style='font-weight: bold;'>" + subject +
"</p>");
```

loadhtml(location, selector)

Retrieve specific content from the filename.

location

String; the location can be absolute or relative to the section/context. Use: snippets/<snippet-name> to retrieve the content from a HTML file residing in snippets folder of the Resources panel.

selector

String. The supplied selector should conform to CSS selector syntax and allows you to retrieve only the content of matching elements.

Examples

This script loads a specific element from the snippet.

```
var mysnippet = loadhtml('snippets/snippet-
selectors.html','#item3');
results.replaceWith(mysnippet);
```

This script loads the children of the selected element.

```
var snippet = loadhtml('snippets/snippet.html','foobar').children
();
results.replaceWith(snippet);
```

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 an JSON enabled server using a standard HTTP request. Popular content management systems, like WordPress (requires JSON API plug-in) and Drupal provide a JSON service/API to retrieve content.

- loadjson(location)

> **Note**
>
> 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

loadjson(location)

Loads json data from a remote location.

location

String; the supplied location should be either a URL or a relative file path.

Examples

This sample script retrieves JSON data from a snippet.

```
var localJSON = loadjson('snippets/jsonsnippet.html');
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)
```

Number functions

- currency()
- currencyNoSymbol()
- grouped()
- integer()
- integerUngrouped()
- number()
- numberUngrouped()

> **Note**
>
> The locale also influences the output of some Number functions; see "Locale" on page 418.

> **Tip**
>
> For fields that contain a number, you can also enter a formatting pattern directly in the Text Script Wizard; see "Using the Text Script Wizard" on page 335, "Formatting variable data" on page 338 and Number patterns).

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 418.

value

A number. This can be a value from a field that contains a SmallInteger, BigInteger, Float, SmallCurrency or LargeCurrency.

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 418. For available patterns, see see Number patterns.

value

A number. This can be a value from a field that contains a SmallInteger, BigInteger, Float, SmallCurrency or LargeCurrency.

pattern

A custom pattern that may consist of symbols; see Number patterns. 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. This can be a value from a field that contains a SmallInteger, BigInteger, Float, SmallCurrency or LargeCurrency.

grouped(value)

Formats a number using a thousands separator. Which separator is used depends on the Locale, see "Locale" on page 418.

value

A number. This can be a value from a field that contains a SmallInteger, BigInteger, Float, SmallCurrency or LargeCurrency.

prepend()

Insert content at the beginning of 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 206). See also: "append()" on page 180.

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 206). `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 <span> 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	<div id="box"> <p>Peter Parker</p> </div>	<div id="box"> <h1>Personal information</h1> <p>Peter Parker</p> </div>

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	<div> <h1>Personal information</h1> <p class="name">Peter Parker</p> </div>	<div> <h1>Personal information</h1> <p class="name"><b>Name: </b>Peter Parker</p> </div>

This script inserts content in multiple <div> elements at the same time.

```
results.prepend("<h1>Personal information</h1>");
```

Selector	Matched element	Matched element after script execution
div	<div id="box"> <p>Peter Parker</p> </div> <div id="box"> <p>Peter Parker</p> </div>	<div id="box"> <h1>Personal information</h1> <p>Peter Parker</p> </div> <div id="box"> <h1>Personal information</h1> <p>Peter Parker</p> </div>

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	<div id="box"> <p>Peter Parker</p> </div>	<div id="box"> <h1>Personal information</h1> <p>Peter Parker</p> </div>

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
<div id="box"> <p>Peter Parker</p> </div>	<div id="box"> <h1>Personal information</h1> <p>Peter Parker</p> </div>

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
<div id="box"> <p>Peter Parker</p> </div>	<div id="box" style="color: red;"> <h1>Personal information</h1> <p>Peter Parker</p> </div>

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");
```

query()

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.

- query(selector)
- query(selector, context)

query(selector)

Creates a new result set containing the HTML elements in the template that match the supplied CSS selector.

selector

A String containing a CSS selector. See http://www.w3schools.com/cssref/css_selectors.asp *for CSS selectors and combinations of CSS selectors.*

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 http://www.w3schools.com/cssref/css_selectors.asp *for CSS selectors and combinations of CSS selectors.*

context

A result set or an HTML string. If the passed context is not a result set of HTML string it will be coerced to a String and interpreted as HTML.

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
<p id="test1">foo</p>	<p id="test1" style="color: yellow;">foo</p>

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);
```

CopyFile()

Examples

removeClass()

Removes the specified class from each element in this result set. 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.addClass("foo");
```

Selector	Matched element	Matched element after script execution
p	<p class="foo">Hello world</p>	<p>Hello world</p>

replaceWith()

Replaces each element in a set of HTML elements.

replaceWith(content)

Replaces 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);
```

Control Script API

When output is generated from a template, Control Scripts run before all other scripts. This topic lists features that can only be used in Control Scripts and provides some sample scripts. See "Control Scripts" on page 385 for more information about how to use this kind of scripts.

Objects available to pre-merge scripts

Channel

This is an enumeration for the output channels. The active output channel is registered in `merge.channel`.

Value	Description
EMAIL	The merge request is for output to Email
PRINT	The merge request is for output to Print
WEB	The merge request is for output to Web
THUMBNAIL	The merge request is for generating a template preview

ContextType

This is an enumeration for the context types.

Value	Description
HTML_EMAIL	The context is the Email context
PRINT	The context is the Print context
WEB	The context is the Web context

Merge

The root level instance of the object `merge` is the entry point in Control Scripts for the objects and references required to query and modify various aspects.

Field	Type	Description
context	Context	The context rendered by this merge pass.
channel	Channel	The channel for which this merge pass is requested.
section	Section	The section being merged. Only available from non-control template scripts AND when the output channel is WEB (merge.channel == Channel.WEB) where it defines the requested page.
template	Template	Template containing the context. This field can be used to find out

		which contexts are available in the template.

Context

Context is an object representing the respective 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 inside this context defined in the template.
type	ContextType	The context type.

Template

Template is an object representing 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
contexts	Array	Array of contexts available in the template.

Section

A section object relating to a section in the template. This type can be used to query and modify the output behavior for when the related context/section is merged.

Field	Type	Description
enabled	boolean	Enables or disables this section for output. Note that even if a 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 (to emulate the

		behavior of previous versions): For **Web** channel requests the requested web section is enabled by default. It is possible to redirect to another section by disabling the requested section and enabling another section. For **Email** channel requests on the **Web** context only the default section is 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 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 **Print** channel requests on the **Print** context all sections are enabled by default.
name	[READONLY] String	The name of the section as shown in the resources view. Note that sections cannot have an integer as name. The name should always include alphanumeric characters. See " Renaming a section" on page 390.
part	String	Name for the part. This is used for channels that can output separate and/or named parts. The Email output, for example, can attach 3 PDF's generated from the Print context. **Part** is used to specify where a new part starts and the title for the part. When used for Email attachments, the part title is used as the file name for the attachment.
restartPageNumber	boolean	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).

BackgroundResource

This is an enumeration for the types of background resources for a Print section.

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.
NONE	No PDF background.
RESOURCE_PDF	A PDF file stored in the template or on the network. Note that it isn't possible to use a remotely stored PDF file as a section's background.

MediaPositon

This is an enumeration for the position of background resources (`background.position`) for a Print section.

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.

Pitfalls to avoid

When using `merge.context` or `merge.section` make sure to check for the appropriate conditions.

- `merge.section` is only defined when when the output channel is Web. To make sure that it is defined, use the following `if` statement: `if (merge.channel == Channel.WEB && merge.context.type == ContextType.WEB) { ... }`.
- 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. Instead of enclosing it in an `if` statement, you can use the template object to access a specific context: `merge.template.contexts.PRINT.sections['Section X']` .

Sample scripts

Conditionally skipping or printing Print sections

```
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

```
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

```
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
- Web sections A and B 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";
} else if (merge.context.type == ContextType.WEB) {
        merge.context.sections['default Section'].enabled = false; //
disable whatever is the default section
        merge.context.sections['Section A'].enabled = true;
        merge.context.sections['Section A'].part = "WebPartA";
        merge.context.sections['Section B'].enabled = true;
        merge.context.sections['Section B'].part = "WebPartB";
}
}
```

Note

For another example, see this how-to: Output sections conditionally.

Setting the background of a Print section

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 rather than using a PDF DataMapper
background
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;
```

Positioning the background of a Print section

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";
```

Basic Steps

With the Designer you can create templates for personalized letters, emails and web pages, 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 420.
2. **Fill the template**
   Add text, images and other elements to the template and style them. See "Content elements" on page 220 and "Styling and formatting" on page 395.

3. **Personalize the content**
   Personalize the content using variable data. See "Personalizing Content" on page 322.
4. **Generate output**
   Adjust the settings, test the template and generate output: letters, emails, and/or web pages. See Skin/Formats/CrossReferencePrintFormat("Generating outputWhen merged with a record set, the templates made in the Designer can generate three types of output: Print, Email and Web.Print outputPrint templates, also called Print sections, are part of the Print context. They are meant to be printed to a printer or printer stream, or to a PDF file (see Generate Print output). The Print context can also be added to Email output as a PDF attachment; see Generating Email output. 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.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.To split the Print output into several files, see Splitting printing into more than one file.Email outputThe 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.Email Output can be generated in two different ways: from the Designer or via Workflow. In both cases, 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. The output of the Web context, as an integral HTML file.Other files, an image or a PDF leaflet for example.Attaching the Print context and/or the Web context is one of the options in the Send (Test) Email dialog; see Generating Email output. To learn how to attach other files, see Email attachments.Web output The Web context outputs an HTML web page that contains the HTML text and all the resources necessary to display it.Web output can be generated in two different ways: it can be attached to an Email template when generating Email output (see above), or it can be generated using Workflow; see Generating Web output.Although the Web context can contain multiple Web pages, only one of them can be merged with each record. Which one is used, depends on a setting; see Web output settings in the Web context and sections." on page 1).
5. **What's next**
   Use Workflow to automate your customer communications.

Note

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:

"Contexts" on the facing page. A context contains one or more designs for one output channel.

"Content elements" on page 220. Elements make up the biggest part of the content of each design.

"Email" on page 294. Email is one of the intended output channels. This topics helps you design an email template.

Skin/Formats/CrossReferencePrintFormat("Generating outputWhen merged with a record set, the templates made in the Designer can generate three types of output: Print, Email and Web.Print outputPrint templates, also called Print sections, are part of the Print context. They are meant to be printed to a printer or printer stream, or to a PDF file (see Generate Print output). The Print context can also be added to Email output as a PDF attachment; see Generating Email output. 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.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.To split the Print output into several files, see Splitting printing into more than one file.Email outputThe 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.Email Output can be generated in two different ways: from the Designer or via Workflow. In both cases, 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. The output of the Web context, as an integral HTML file.Other files, an image or a PDF leaflet for example.Attaching the Print context and/or the Web context is one of the options in the Send (Test) Email dialog; see Generating Email output. To learn how to attach other files, see Email attachments.Web output The Web context outputs an HTML web page that contains the HTML text and all the resources necessary to display it.Web output can be generated in two different ways: it can be attached to an Email template when generating Email output (see above), or it can be generated using Workflow; see Generating Web output.Although the Web context can contain multiple Web pages, only one of them can be merged with each record. Which one is used, depends on a setting; see Web output settings in the Web context and sections." on page 1). Learn the ins and outs of generating output from each of the contexts.

"Personalizing Content" on page 322. Personalize your customer communications using variable data.

"Print" on page 348. Print is one of the intended output channels. This topic helps you design and fill sections in the Print context.

"Sections" on page 389. Sections in one context are designed for the same output channel.

"Snippets" on page 393. Snippets help share content between contexts, or insert content conditionally.

"Styling and formatting" on page 395. Make your Designer templates look pretty and give them the same look and feel with style sheets.

"Templates" on page 420. Start creating, using and sharing templates.

"Web" on page 441. Web is one of the intended output channels. This topic helps you design a web page.

"Write your own scripts" on page 373. Scripting can take personalization much further. Learn how to script via this topic.

Contexts

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

- The Print context outputs documents to either a physical printer a PDF file; see "Print context" on page 349.
- The Email context outputs HTML email, composed of HTML code with embedded CSS. See "Email context" on page 295.
- The Web context outputs an HTML web page. See "Web Context" on page 441.

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" below.

Tip

If an Email context is going to be part of the template, it is recommended to start with an Email Template Wizard; see "Creating an Email template with a Wizard" on page 425. After creating a template, contexts can be added to it, but that can not be done with a wizard.

Outputting and combining contexts

All three contexts can be present in any template and they can all be used to output documents; see "Generating Email output" on page 317, "Generating Print output" on page 306 and "Generating Web output" on page 320.

They can even be combined in output.

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

Outputting other combinations of contexts, and selecting sections based on a value in the data, can be done via a Control Script; see "Control Scripts" on page 385.

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**. Only one context of each type can be present in a template. Each context, however, can hold more than one section; see "Sections" on page 389.

Deleting a context

To delete a context, right-click the context on the **Resources** pane and 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. After closing and reopening the template it is no longer possible to restore the deleted context this way.

Content elements

Once you have created template, it can be filled with all kinds of elements, from text to barcodes and from tables to fields on a web form. All types of elements are listed on this page; see below.

There are several ways to insert elements, see "Inserting an element" on page 223.

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 224, "Attributes" on page 222 and "Styling and formatting an element" on page 224.

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 the next page.

Element types

The following types of content can be added to the content of a template:

- "Images" on page 276 and "Dynamic Images" on page 343
- "Text and special characters" on page 282
- "Date" on page 264
- "Table" on page 280and "Dynamic table" on page 344
- "Boxes" on page 262: Positioned Box, Inline Box, Div and Span

> **Tip**
>
> Wrapping elements in a box (see "Boxes" on page 262) or in a semantic HTML element makes it easier to target them in a script or 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 335 and "Styling and formatting" on page 395.

- "Hyperlink and mailto link" on page 274
- "Barcode" on page 225
- Web "Forms" on page 266 and Web "Form Elements" on page 269
- "COTG Elements" on page 290
- "Whitespace elements: using optional space at the end of the last page" on page 361 (Print context only)
- "Page numbers" on page 362 (Print context only)
- Article, Section, Header, Footer, Nav and Aside are HTML5 semantic elements; see http://www.w3schools.com/html/html5_semantic_elements.asp
- Other HTML elements: Heading, Address and Pre
- "Snippets" on page 393: 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. Forms and Form elements can be used on web pages only, whereas 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 and Web 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 407 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. 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 322) and styling (see "Styling templates with CSS files" on page 396).

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 224) and type an ID and/or a class in the respective fields on the **Attributes** pane at the top right.

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.

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 335.

In code, you can change an element's attribute using the function `attr()`; see "Write your own scripts" on page 373 and "API" on page 171.

Inserting an element

To insert an element in the content of a template:

1. Click the respective toolbar button. Alternatively, click the element on the **Insert** menu.
2. Add an ID and/or a class. ID's and classes are particularly useful with regard to variable data (see "Personalizing Content" on page 322) and styling (see "Styling templates with CSS files" on page 396).
3. Use the **Location** drop-down to select where to insert the element.
    - **At cursor position**: The element is inserted where the cursor is located in the template.
    - **Before element**: The element is inserted before the current HTML element where the cursor is located. For example if the cursor is within a paragraph, insertion occurs before the <p> tag.*
    - **After start tag**: The element is inserted within the current HTML element, at the beginning, just after the start tag.*
    - **Before end tag**: The element is inserted within the current HTML element, at the end, just before the end tag.*
    - **After element**: The element is inserted after the current element where the cursor is located. For example if the cursor is within a paragraph, insertion occurs after the end tag of the paragraph (</p>).*

    * 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 until the root of the body.

Selecting an element

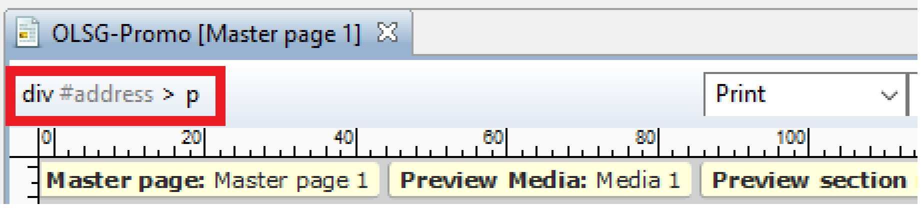
When an element is selected, the **Attributes** pane shows the attributes of that element, and the **Styles** pane, next to the **Attributes** pane, shows which styles are applied to it.

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 to make borders of elements visible on the Design tab. The borders will not be visible on the Preview tab.

There are two more ways to select an element in the content:

- Use 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`. 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.
- Use 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.

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 395 for more information about the formatting options.

Format elements via Cascading Style Sheets (CSS)

It is recommended to use style sheets in your templates right from the start, even more so if your communications are going to be output to different output channels, or if they consist out of different sections (for example, a covering letter and a policy). With CSS you can give your templates one look and feel. 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, for example. See "Styling templates with CSS files" on page 396.

Barcode

In PReS 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 Text" on page 540) 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 Settings" on page 583) and send the preset to Workflow.

Before adding a Barcode, load data or at least a Data Model; see "Loading data" on page 324. 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. An **ID** is required. You can change the given ID and, optionally, add a class.
4. Use the **Location** drop-down to select where to insert the Barcode.
    - **At cursor position** inserts the Barcode 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 Barcode is inserted before the <p> 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 Barcode is inserted after the end tag of the paragraph (</p>).*
    - **Absolute on body** inserts the Barcode in an absolute-positioned box inside the <body> of the HTML, but outside other elements.

   * 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 until the root of the body.
5. Under **Script**, select the field that contains the barcode value, or select the fields that together compose 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. The barcode type dictates the length and exact format of the required value.
   For barcodes that require a Checksum, the Designer can calculate a Checksum if that isn't provided by your data. Then script result should be the required value minus the Checksum. Edit the barcode properties to include the calculated Checksum with the barcode value, after adding the barcode to the template; see below.

> **Note**
>
> 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.

6. Click **OK** to close the dialog.

In the template the barcode shows up as a gray box. The barcode script is added to the Scripts pane. To see the barcode script working, toggle to the **Preview** tab in the Workspace.

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(s) are added to the barcode value.

Barcode properties

A barcode is always added with 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 respective barcode on the shortcut menu.

Click the barcode type below for information about its properties.

- "Code 11, Code 93, Code 93 extended, Industrial 2 of 5, Interleaved 2 of 5, Matrix 2 of 5" on page 248
- "Code 39, Code 39 extended" on page 234
- "UPC-A, UPC-E, EAN-8, EAN-13" on page 260
- "OneCode, KIX Code, Royal Mail, Australia Post" on page 258
- "Code 128" on page 237
- "GS1-128" on page 242
- "Codabar" on page 231
- "MSI" on page 250
- "IMPB" on page 244
- "Postnet" on page 254
- "QR Code" on page 256
- "Data Matrix" on page 238
- "Royal Mail Mailmark" on page 259

- "PDF417" on page 252
- "Aztec Code" on the next page
- "MaxiCode" on page 249

> **Tip**
>
> As of version 1.5 it will be possible to change the type of a barcode in the template as well, when changing the properties of a barcode.

OneCode, KIX Code, Royal Mail, Australia Post

OneCode, KIX Code, Royal Mail and Australia Post are some of the types of barcodes that can be added to a template; see "Barcode" on page 225.

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 225.

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 type on the shortcut menu.

Barcode properties

This topic lists the properties of the barcode types OneCode, KIX Code, Royal Mail and Australia Post. For the properties of other barcode types, see "Barcode properties" on the previous page.

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.
- **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.

Color

The **Color** property 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).

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is smaller in size, but not compatible with Email output.
- **PNG**: Binary rasterized format. This is slightly larger than SVG but will display properly in Email output.

Aztec Code

Aztec is one of the types of barcodes that can be added to a template; see "Barcode" on page 225.

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 225.

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 type on the shortcut menu.

Barcode properties

This topic lists the properties of the barcode type Aztec. For the properties of other barcode types, see "Barcode properties" on page 227.

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.
- **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.

Color

The **Color** property 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).

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is smaller in size, but not compatible with Email output.
- **PNG**: Binary rasterized format. This is slightly larger than SVG but 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 225.

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 type on the shortcut menu.

Barcode properties

This topic lists the properties of the Codabar barcode. For the properties of other barcode types, see "Barcode properties" on page 227.

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.
- **Proportionally**: 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, PReS 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, PReS 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** property 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).

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is smaller in size, but not compatible with Email output.
- **PNG**: Binary rasterized format. This is slightly larger than SVG but will display properly in Email output.

Code 11, Code 93, Code 93 extended, Industrial 2 of 5, Interleaved 2 of 5, Matrix 2 of 5

Code 11, Code 93, Code 93 extended, 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 225.

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 type on the shortcut menu.

Barcode properties

This topic lists the properties of the following barcode types :

- Code 11
- Code 93
- Code 93 extended
- Industrial 2 of 5
- Interleaved 2 of 5
- Matrix 2 of 5

For the properties of other barcode types, see "Barcode properties" on page 227.

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.
- **Proportionally**: 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, PReS 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, PReS 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** property 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).

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is smaller in size, but not compatible with Email output.
- **PNG**: Binary rasterized format. This is slightly larger than SVG but will display properly in Email output.

Code 39, Code 39 extended

Code 39 and Code 39 extended are two 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 225.

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 type on the shortcut menu.

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 properties" on page 227.

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.
- **Proportionally**: 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, PReS 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, PReS 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** property 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).

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is smaller in size, but not compatible with Email output.
- **PNG**: Binary rasterized format. This is slightly larger than SVG but will display properly in Email output.

Code 11, Code 93, Code 93 extended, Industrial 2 of 5, Interleaved 2 of 5, Matrix 2 of 5

Code 11, Code 93, Code 93 extended, 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 225.

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 type on the shortcut menu.

Barcode properties

This topic lists the properties of the following barcode types :

- Code 11
- Code 93
- Code 93 extended
- Industrial 2 of 5

- Interleaved 2 of 5
- Matrix 2 of 5

For the properties of other barcode types, see "Barcode properties" on page 227.

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.
- **Proportionally**: 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, PReS 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, PReS 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** property 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).

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is smaller in size, but not compatible with Email output.
- **PNG**: Binary rasterized format. This is slightly larger than SVG but 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 225.

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 225.

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 type on the shortcut menu.

Barcode properties

This topic lists the properties of the barcode type Code 128. For the properties of other barcode types, see "Barcode properties" on page 227.

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.

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.

Add Checksum

When checked, PReS 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, PReS 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** property 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).

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is smaller in size, but not compatible with Email output.
- **PNG**: Binary rasterized format. This is slightly larger than SVG but 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 225.

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 225.

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 type on the shortcut menu.

Barcode properties

This topic lists the properties of the QR barcode. For the properties of other barcode types, see "Barcode properties" on page 227.

Dots per pixel

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.
- **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.

Color

The **Color** property 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).

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is smaller in size, but not compatible with Email output.
- **PNG**: Binary rasterized format. This is slightly larger than SVG but 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 225.

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 type on the shortcut menu.

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 properties" on page 227.

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.
- **Proportionally**: The barcode is stretched up to where it fits either the width or height of the parent box, whichever requires the less stretching.

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, PReS 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** property 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).

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is smaller in size, but not compatible with Email output.
- **PNG**: Binary rasterized format. This is slightly larger than SVG but 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 225.

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 225.

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 type on the shortcut menu.

Barcode properties

This topic lists the properties of the barcode type GS1-128. For the properties of other barcode types, see "Barcode properties" on page 227.

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.
- **Proportionally**: 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, PReS 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, PReS 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** property 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).

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is smaller in size, but not compatible with Email output.
- **PNG**: Binary rasterized format. This is slightly larger than SVG but will display properly in Email output.

OneCode, KIX Code, Royal Mail, Australia Post

OneCode, KIX Code, Royal Mail and Australia Post are some of the types of barcodes that can be added to a template; see "Barcode" on page 225.

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 225.

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 type on the shortcut menu.

Barcode properties

This topic lists the properties of the barcode types OneCode, KIX Code, Royal Mail and Australia Post. For the properties of other barcode types, see "Barcode properties" on page 227.

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.
- **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.

Color

The **Color** property 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).

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is smaller in size, but not compatible with Email output.
- **PNG**: Binary rasterized format. This is slightly larger than SVG but will display properly in Email output.

IMPB

IMPB is one of the barcode types that can be added to a template; see "Barcode" on page 225.

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 225.

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 type on the shortcut menu.

Barcode properties

This topic lists the properties of the barcode type IMPB. For the properties of other barcode types, see "Barcode properties" on page 227.

Color

The **Color** property 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).

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is smaller in size, but not compatible with Email output.
- **PNG**: Binary rasterized format. This is slightly larger than SVG but will display properly in Email output.

Code 11, Code 93, Code 93 extended, Industrial 2 of 5, Interleaved 2 of 5, Matrix 2 of 5

Code 11, Code 93, Code 93 extended, 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 225.

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 type on the shortcut menu.

Barcode properties

This topic lists the properties of the following barcode types :

- Code 11
- Code 93
- Code 93 extended
- Industrial 2 of 5
- Interleaved 2 of 5
- Matrix 2 of 5

For the properties of other barcode types, see "Barcode properties" on page 227.

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.
- **Proportionally**: 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, PReS 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, PReS 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** property 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).

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is smaller in size, but not compatible with Email output.
- **PNG**: Binary rasterized format. This is slightly larger than SVG but will display properly in Email output.

Code 11, Code 93, Code 93 extended, Industrial 2 of 5, Interleaved 2 of 5, Matrix 2 of 5

Code 11, Code 93, Code 93 extended, 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 225.

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 type on the shortcut menu.

Barcode properties

This topic lists the properties of the following barcode types :

- Code 11
- Code 93
- Code 93 extended
- Industrial 2 of 5
- Interleaved 2 of 5
- Matrix 2 of 5

For the properties of other barcode types, see "Barcode properties" on page 227.

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.
- **Proportionally**: 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, PReS 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, PReS 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** property 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).

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is smaller in size, but not compatible with Email output.
- **PNG**: Binary rasterized format. This is slightly larger than SVG but will display properly in Email output.

Code 11, Code 93, Code 93 extended, Industrial 2 of 5, Interleaved 2 of 5, Matrix 2 of 5

Code 11, Code 93, Code 93 extended, 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 225.

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 type on the shortcut menu.

Barcode properties

This topic lists the properties of the following barcode types :

- Code 11
- Code 93
- Code 93 extended
- Industrial 2 of 5
- Interleaved 2 of 5
- Matrix 2 of 5

For the properties of other barcode types, see "Barcode properties" on page 227.

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.
- **Proportionally**: 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, PReS 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, PReS 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** property 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).

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is smaller in size, but not compatible with Email output.
- **PNG**: Binary rasterized format. This is slightly larger than SVG but 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 225.

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 225.

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 type on the shortcut menu.

Barcode properties

This topic lists the properties of the MaxiCode barcode. For the properties of other barcode types, see "Barcode properties" on page 227.

Resolution

Select the printer output definition for the barcode (200, 300, 400, 500 or 600 dpi).

Color

The **Color** property 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).

MSI

MSI is one of the types of barcodes that can be added to a template; see "Barcode" on page 225.

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 225.

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 type on the shortcut menu.

Barcode properties

This topic lists the properties of the barcode type MSI. For the properties of other barcode types, see "Barcode properties" on page 227.

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.
- **Proportionally**: 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, PReS 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, PReS 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** property 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).

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is smaller in size, but not compatible with Email output.
- **PNG**: Binary rasterized format. This is slightly larger than SVG but will display properly in Email output.

OneCode, KIX Code, Royal Mail, Australia Post

OneCode, KIX Code, Royal Mail and Australia Post are some of the types of barcodes that can be added to a template; see "Barcode" on page 225.

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 225.

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 type on the shortcut menu.

Barcode properties

This topic lists the properties of the barcode types OneCode, KIX Code, Royal Mail and Australia Post. For the properties of other barcode types, see "Barcode properties" on page 227.

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.
- **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.

Color

The **Color** property 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).

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is smaller in size, but not compatible with Email output.
- **PNG**: Binary rasterized format. This is slightly larger than SVG but 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 225.

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 225.

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 type on the shortcut menu.

Barcode properties

This topic lists the properties of the barcode type PDF417. For the properties of other barcode types, see "Barcode properties" on page 227.

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.

Color

The **Color** property 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).

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is smaller in size, but not compatible with Email output.
- **PNG**: Binary rasterized format. This is slightly larger than SVG but 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 225.

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 225.

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 type on the shortcut menu.

Barcode properties

This topic lists the properties of the barcode type Postnet. For the properties of other barcode types, see "Barcode properties" on page 227.

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, PReS 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** property 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).

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is smaller in size, but not compatible with Email output.
- **PNG**: Binary rasterized format. This is slightly larger than SVG but 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 225.

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 225.

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 type on the shortcut menu.

Barcode properties

This topic lists the properties of the QR barcode. For the properties of other barcode types, see "Barcode properties" on page 227.

Module size

Enter the size of the square modules in pixels.

Auto configure

When this option is checked, the barcode generator overwrites the selected Preferred version (see below) and defines the barcode version based on the supplied data.

Preferred version

There are 40 sizes of QR codes. Select the preferred version for the QR code.

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**, **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:

- **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.

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.

Color

The **Color** property 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).

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is smaller in size, but not compatible with Email output.
- **PNG**: Binary rasterized format. This is slightly larger than SVG but 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.

OneCode, KIX Code, Royal Mail, Australia Post

OneCode, KIX Code, Royal Mail and Australia Post are some of the types of barcodes that can be added to a template; see "Barcode" on page 225.

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 225.

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 type on the shortcut menu.

Barcode properties

This topic lists the properties of the barcode types OneCode, KIX Code, Royal Mail and Australia Post. For the properties of other barcode types, see "Barcode properties" on page 227.

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.
- **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.

Color

The **Color** property 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).

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is smaller in size, but not compatible with Email output.
- **PNG**: Binary rasterized format. This is slightly larger than SVG but will display properly in Email output.

Royal Mail Mailmark

Royal Mail Mailmark is one of the types of barcodes that can be added to a template; see "Barcode" on page 225.

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 225.

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 type on the shortcut menu.

Barcode properties

This topic lists the properties of the barcode type Royal Mail Mailmark. For the properties of other barcode types, see "Barcode properties" on page 227.

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.
- **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.

Color

The **Color** property 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).

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is smaller in size, but not compatible with Email output.
- **PNG**: Binary rasterized format. This is slightly larger than SVG but 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 225.

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 type on the shortcut menu.

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 properties" on page 227.

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.
- **Proportionally**: The barcode is stretched up to where it fits either the width or height of the parent box, whichever requires the less stretching.

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, PReS 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** property 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).

Output format

Defines how the barcode is output on the page. There are two possible formats:

- **SVG**: Vector format. This is smaller in size, but not compatible with Email output.
- **PNG**: Binary rasterized format. This is slightly larger than SVG but will display properly in Email output.

Boxes

Boxes are elements that are used to surround other elements, either to style them or to place them in specific locations.

> **Tip**
>
> Wrapping elements in a box (see "Boxes" above) or in a semantic HTML element makes it easier to target them in a script or 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 335 and "Styling and formatting" on page 395.

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 <div> element that has an **absolute position**; in other words, it has its CSS property `position` set to `absolute`.

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. Positioned Boxes can be moved by dragging the borders, and resized using the handles on the sides and the corners. 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 395 and "Styling templates with CSS files" on page 396.

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 <div> 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 395 and "Styling templates with CSS files" on page 396.

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 <div> 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 (<span> 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 335 and "Styling and formatting" on page 395.

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 395 and "Styling templates with CSS files" on page 396.

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 223.

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 221.

In HTML, boxes are <div> elements. Spans are <span> elements. To learn how to change the attributes of elements, see "Attributes" on page 222.

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, use the **Insert > Date** option in the "Menus" on page 479. 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.

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 http://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 197).
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 189.

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; see "Locale" on page 418.

Forms

Web templates can contain Form elements. Capture OnTheGo templates always contain a Form element.

> **Tip**
>
> To create a Capture OnTheGo template, preferably use a Template Wizard; see "Capture OnTheGo (COTG) templates" on page 433. The Wizard not only adds the form, but also the necessary Capture OnTheGo form elements, styles and JavaScript files.

Adding a Form

1. On the **Resources** pane, expand the **Web** context and double-click a Web page to open it.
2. To use the Form Wizard, select the **Insert > Form Wizard** menu option. The Form Wizard adds a Form to the Web page including the specified fields.
   Alternatively, you can select **Insert > Form** on the menu to open a dialog that lets you set the Form's properties, validation method and location, but doesn't allow you to specify fields. If you choose this method, skip step 8 and 9 of this procedure and add fields after inserting the Form (see "Forms" above).
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 322) and styling (see "Styling templates with CSS files" on page 396).
4. In the **Action** field, enter the URL where the form data should be sent. The URL should be a server-side script that can accept form data. The **action** of a Capture OnTheGo form should specify the Workflow HTTP Server Input task that receives and handles the submitted data. The action will look like this: **http://127.0.0.1:8080/action** (8080 is Workflow's default port number; 'action' should be replaced by the HTTP action of that particular HTTP Server Input task).
   The **method** of a Capture OnTheGo form should be **POST** to ensure that it doesn't hit a data limit when submitting the form. The **GET** method adds the data to the URL, and the length of a URL is limited to 2048 characters.

5. Using the the **Method** drop-down, select whether the form should be sent using the GET or POST method.
6. Using the next drop-down, select the form's Encryption Type (**enctype**):
    - **application/x-www-form-urlencoded**: Default. All characters are encoded before they are sent. Spaces are converted to "+" symbols, and special characters are converted to ASCII HEX values.
    - **multipart/form-data**: No characters are encoded. This value is required when you are using forms that have a file upload control.
    - **text/plain**: Spaces are converted to "+" symbols, but no special characters are encoded.
7. Select a **validation** method:
    - The **Browser** validation method leaves it up to the browser to validate the user input. When adding fields to the Form (see the next step) you can only make fields required and set the maximum length as an additional requirement for some fields.
    - Select **JQuery Validation** to validate using JQuery scripts. This allows you to specify stricter requirements per field and type a different message for each field to display to the user if the input is not valid. This method ensures a more consistent validation as it is browser independent. The necessary JQuery files will be added to the JavaScript folder on the Resources pane when the form is inserted.
8. Under **Fields**, click the **Add** button and click on the desired field type to add a field of that type; see "Form Elements" on page 269.

> **Note**
>
> A Fieldset is not available in the Form Wizard, because a Fieldset itself can contain multiple different fields. Add the Fieldset after inserting the Form; see "Forms" on the previous page.

9. Double-click each field in the Fields list and change its settings. For an explanation of the settings, see "Forms" on the previous page.
10. The order of the elements in the list under **Fields** determines in which order the elements will be added to the Form. Use the **Move Up** and **Move Down** buttons to change the order of the elements in the list.

11. Use the **Location** drop-down to select where to insert the element.

    - **At cursor position**: The element is inserted where the cursor is located in the template.
    - **Before element**: The element is inserted before the current HTML element where the cursor is located. For example if the cursor is within a paragraph, insertion occurs before the <p> tag.*
    - **After start tag**: The element is inserted within the current HTML element, at the beginning, just after the start tag.*
    - **Before end tag**: The element is inserted within the current HTML element, at the end, just before the end tag.*
    - **After element**: The element is inserted after the current element where the cursor is located. For example if the cursor is within a paragraph, insertion occurs after the end tag of the paragraph (</p>).*

    * 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 (at the top of the Workspace), from the current selection point until the root of the body.
12. Close the dialog.

Modifying a Form

To modify a Form, you can of course edit the HTML code directly in the Source view of the Workspace. Apart from that, there are a number of dialogs to change a Form's properties, validation settings and the elements inside the Form.

Changing a Form's properties

1. Select the form (see "Selecting an element" on page 224).
2. On the **Attributes** pane you can change:

    - The **ID** and/or **class**. ID's and classes are particularly useful with regard to variable data (see "Personalizing Content" on page 322) and styling (see "Styling templates with CSS files" on page 396).
    - An **Action**: the URL where the form data should be sent. The URL should be a server-side script that can accept form data.
    - A **Method**: this defines whether the form should be sent using the GET or POST method.
    - An Encryption Type (**enctype**):
        - **application/x-www-form-urlencoded**: Default. All characters are encoded before they are sent. Spaces are converted to "+" symbols, and special

characters are converted to ASCII HEX values.
  - **multipart/form-data**: No characters are encoded. This value is required when you are using forms that have a file upload control.
  - **text/plain**: Spaces are converted to "+" symbols, but no special characters are encoded.

Changing a Form's validation method

In Connect PReS Connect, there are two ways in which a Form's input can be validated:

- The **Browser** validation method leaves it up to the browser to validate the user input. When adding fields to the Form (see the next step) you can only make fields required and set the maximum length as an additional requirement for some fields.
- Select **JQuery Validation** to validate using JQuery scripts. This allows you to specify stricter requirements per field and type a different message for each field to display to the user if the input is not valid. This method ensures a more consistent validation as it is browser independent. The necessary JQuery files will be added to the JavaScript folder on the Resources pane when the form is inserted.

To change a Form's validation method:

1. **Right-click** on the Form element and choose **Validation settings**.
2. Choose a validation type (see above).
3. Double-click each field in the list to edit their validation settings:
   - **Required**: Check if the field is required to submit the form. If a field is required but contains no data, a message will be shown to the user.
   - **Minimum length**: Enter a numerical value for the minimum character length required for this field.
   - **Maximum length**: Enter a numerical value for the minimum character length accepted for this field.
   - **Equal to**: Use the drop-down to select another field that is already added to the same Form. The contents of both fields must match for the data to be validated. This is useful for confirmation fields such as for passwords, email addresses etc.

   Which of these options are available depends on the validation method of the form: with Browser validation you can only make a field required and set a maximum length.

Form Elements

This topic lists the elements that can be added to a form in a Web page or in a Capture OnTheGo template and explains how to add them to a Form, set a default value or change their validation. For more information about Forms, see "Forms" on page 266. For more information about elements in Forms, see http://www.w3schools.com/html/html_forms.asp.

Types of Form Elements

Fieldset

A fieldset is a group of related elements in a form. The elements don't have to be of the same type. After inserting and selecting the Fieldset (see "Selecting an element" on page 224) you can add elements to it in the same way you add elements to a Form; see "Adding elements to a Form" on page 272.

Text

The Text element is a simple <input> element with the type `text`. It accepts any alphanumerical characters, including special characters. The string is HTML-encoded before it is submitted to the server.

Email

The Email element is a text input field that accepts only email addresses in a valid format.

URL

The URL element is a text input field that accepts only URLs (a web page address) in a valid format.

Password

The Password element is a password field that accepts any alphanumerical characters. When the user types in this field, characters are shown as asterisks only.

Number

The Number element is a text field accepting only numbers.

Date

The Date element is a text field accepting only dates in a valid format.

Text area

The Text area element is a text field accepting multiple paragraphs. You can set a number of rows when adding the Text Area to the Form, and change it on the Attributes pane.

Hidden field

A hidden field can contain specific data used by the server-side script. It is not visible to the user. When adding a Hidden Field you can set the value that will be sent on submit.

Label

A Label element is a text displaying informative text within the form. Labels are non-interactive. Note that this type of label is not tied to an input element. At the same time you add an input element, you can add a label to that element; see "Adding elements to a Form" on the facing page below. To change a label after inserting the Form, simply click it and start typing.

Checkbox

A Checkbox element sends information to the server whether it is checked (true) or not (false). When adding a Checkbox you can set its initial state and its value. The contents of the value property do not appear in the user interface. If a Checkbox is in checked state when the form is submitted, the `name` of the checkbox is sent along with its value (if the Checkbox is not checked, no information is sent).

Radio Button Group

A Radio Button Group is not an input element itself. Rather it is a group of Radio Buttons that have the same `submit name`, indicating that they belong to the same group. Multiple Radio Buttons in the same group only accept one option to be selected. Only the value of the selected Radio Button will be sent to the server on submitting the form. If more options are to be allowed at the same time you should use Checkboxes instead.
The option to add a Radio Button Group is only available in the Form Wizard; see "Forms" on page 266. You could also create a Radio Button Group by specifying the same submit name for a number of Radio Buttons when adding them to a Form.

Radio Button

A radio button sends information to the server whether it is selected (true) or not (false). When adding a Radio Button you can set its initial state and its value. The contents of the value property do not appear in the user interface. If a Radio Button is in selected state when the form is submitted, the `name` of the Radio Button is sent along with its value (if it is not selected, no information is sent).

The `submit name` of a Radio Button indicates to which Radio Button Group the Radio Button belongs.

Select

A Select element is a drop-down list with multiple entries from which the user can select only one option. When adding a Select to a Form you can type the options to be given in the drop-down list and the values related to them. Only the value of the selected option will be sent to the server on submitting the form.

Button

The Button element is an element that can be clicked. When adding a Button to a Form you can set the button's **type**:

- A **submit** button will validate the form data and if validation is successful, send the data to the provided URL (by the Action specified for the Form; see "Changing a Form's properties" on page 268).
- A **reset** button will reset the form to its original configuration, erasing any information entered and options provided. **Note**: This cannot be undone!

The button's type can also be changed on the Attributes pane.

There may be multiple submit buttons on a Form. If there are, specify a different `name` and/or `value` for each of the buttons. When adding a Button to a Form, you can specify a value, but no name. The Button's ID will be copied to the element's `name` attribute. However, after inserting the Button you can change the `name` on the Attributes pane.

Adding elements to a Form

To add an element to a Form or Fieldset, click inside the Form or Fieldset, select **Insert > Form elements**, and choose the respective element on the menu. Now you can change the element's settings:

1. Add an **ID** (required) and, optionally, a **class**.

   > **Note**
   >
   > The ID will be copied to the `name` attribute of the element. The `name` attribute is what identifies the field to the receiving server-side script. To change the name, select the element after inserting it and type the new name on the **Attributes** pane.

   ID's and classes are also useful with regard to variable data (see "Personalizing Content" on page 322) and styling (see "Styling templates with CSS files" on page 396).
2. Type a label, or choose No label under Style, to omit the label. (For Label elements there are no other options to be set.)
3. If applicable, choose a style for the label (for the label of a Checkbox, for example, you can't set a style).

- **Wrap input with label** places the input element inside the Label element.
- **Attach label to input** ties the label to the input element using the `for` attribute of the Label element.
- **Use label as placeholder** inserts the given label text in the placeholder attribute of the field.
- **No style** omits the label altogether.

> **Note**
>
> The first two label styles ensure that when the user clicks the label, the input element gets the focus.

4. The following options are only available for specific elements:
   - For a **Text Area** you can specify a **number of rows**.
   - For a **Radio Button**, the **submit name** indicates to which Radio Button Group the Radio Button belongs.
   - For a **Button**, **Checkbox**, **Hidden Field**, and **Radio Button** you can set the **value**. The value is associated with the input and will be sent on submitting the Form.

   > **Tip**
   >
   > For other Form elements, you can set the **default value** to be the value of a field in the record set; see "Specifying a default value" on the facing page.

   - For a **Checkbox** or **Radio Button** you can check **checked** or **selected** respectively for the element to initially be checked/selected when the web page is shown.
   - For a **Button**, you can set the **button type**:
     - **Submit**: The button will validate the form data and if validation is successful, send the data to the provided URL (the action specified for the Form; see "Changing a Form's properties" on page 268).
     - **Reset**: The button will reset the form to its original configuration, erasing any information entered and options provided. **Note**: This cannot be undone!
5. Depending on the validation method of the form (see "Form Elements" on page 269) and the type of element there are a number of options to set under **Validation**. With Browser validation you can only make a field required and set a maximum length.

- **Required**: Check if the field is required to submit the form. If a field is required but contains no data, a message will be shown to the user.
- **Minimum and maximum length**: Enter a numerical value for the minimum and maximum character length required for this field.
- **Equal to**: Use the drop-down to select another field that is already added to the same Form. The contents of both fields must match for the data to be validated. This is useful for confirmation fields such as for passwords, email addresses etc.

6. Under **Warnings**, type the message that will be displayed to the user if the input is not valid.

When adding an element to a Form or Fieldset, you cannot specify a `name`; the ID will be copied to the element's `name` attribute. After adding the element to the Form or Fieldset you can change the `name` on the Attributes pane. The `name` attribute of Form elements is sent to the server after the form has been submitted.

Specifying a default value

Attribute a default value to a Text, Textarea and other Form elements by dragging a field from the Data Model pane directly onto the field, once it has been created. This also works when dragging a field from a detail table in a record set into a Form element that is contained within a Dynamic Table.
Note that the default value doesn't disappear when the user clicks the field, as placeholders do. To insert a placeholder in a field, type a label and choose **Use label as placeholder** as its style when adding the element to the form; see "Forms" on page 266.

Changing the validation

To change the validation of a Form element, right-click the element and choose **Validation settings**. Now you can change the Form's validation method and set the requirements per field; see "Changing a Form's validation method" on page 269.

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.

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 221.

The HTML tag of a hyperlink or mailto-link is <a>. This is sometimes called an `anchor` tag. For a list of attributes, see http://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 a regular hyperlink pointing to a web page, or select Email to create a mailto-link that will open the default email client when clicked.
4. For a **URL**:
    - **URL**: enter a valid, well-formed URL to link to. It must start with the protocol, such as http:// or https://.
    - **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.

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.

How to add or modify a hyperlink is described in a how-to; see How to dynamically insert a hyperlink. This implies writing a script. For information about scripts, see "Write your own scripts" on page 373.

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" on page 347.

Images

Images are a powerful ingredient in all of your templates. This topic explains how to add and use them.

Ways to use images

In templates, both **imported** images and **external** images can be used (see "Adding images" on the next page and "Resources" on page 439). Once added to the content of a template, an image can be resized (see "Styling an image" on page 278) and alternate text can be linked to it (see "Setting an alternate text" on page 279).

Dynamic image

Images can be switched dynamically, so that a letter, email or web page can include one image or another, depending on a value in the data set. Read "Dynamic Images" on page 343 to find out how to add such switching images.

Background image

Several parts of templates can have a background image. See "Background color and/or image" on page 409 and "Using a PDF file as background image" on page 356.

> **Tip**
>
> To create a Print section from an existing PDF file, use a PDF file as a Print section's background. 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. See "Using a PDF file as background image" on page 356.

Optional filler for 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, use an image or advert as a 'whitespace element'; see "Images" on the previous page.

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 221.

In HTML, 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 279.

The value of the attributes can be changed via a script; see "Attributes" on page 222.

Adding images

To add an image to a template:

1. Select **Insert > Image > From file**, or click the **Insert Image** button on the toolbar. A standard Open File dialog appears.
2. Browse to the file and click **Open**.
3. Check the option **Save with template** to import the image. Imported images are saved within the template file, so they are always available and don't depend on external paths, servers or URLs. If this option is checked, the image file will be inserted in the **Images** folder on the **Resources** pane at the top left.
   Uncheck this option if the image should be used as an external image (see also: "Using external images" on the facing page).
4. Click **Finish**. The image will be inserted at the position of the cursor.

Alternatively:

1. Look up the image file on your computer and drag the image file from the Explorer to the **Images** folder on the **Resources** pane at the top left.
2. To place the image in the content, drag it from the **Images** folder on the **Resources** pane to the content and drop it.

To import a remotely stored image, see "Using external images" below.

Using external images

An external image is an image that is not saved within the template file. It is located either on a specific website (URL) or on a hard drive that is accessible from the machine on which the template's output is produced.

When external images are used, they are updated (retrieved) when the template is merged with a record set to generate output.

To use an external image:

1. Position the cursor in the template where you want the image to be inserted.
2. On the menu, select **Insert > Image > From address**, or on the toolbar, click the **Insert Image from address** button. The Insert Image from Address dialog appears.
3. Enter the **URL**. This can either be
   - a web address (http://www.mysite.com/images/image.jpg), or
   - a local path, for example: file:///c:/resources/images/image.jpg. 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'.
4. Enter an **alternate text**. This text will be shown in emails and on web pages at the position of the image while the image is loading and when the image is not found. On web pages, alternate texts are also used for accessibility.
5. Click **Finish**. The image will be inserted at the current position of the cursor in the template.

For information about referring to images in HTML or in a script, see "Resources" on page 439.

Styling an image

Images can be styled using the **Format > Image** menu item. They have a border, margin and padding; see "Border" on page 410 and "Spacing" on page 419. Images can also be left-aligned or right-aligned, just like boxes.

Resizing an image

There are three 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 224) 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.

Positioning an image

Initially an image will float to the left. Select the image (see "Selecting an element" on page 224) and use the (Float left), (No float) 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.

To position an image using the menu, select the image and then select one of the options in **Format > Float**.

Alternatively, select the image and on the menu, select **Format > Image** and on the **Image** tab, set the **Float** property. This is equivalent to the `float` property in CSS.

Rotating an image

To rotate an image, select the image (see "Selecting an element" on page 224) and select **Format > Image**, on the menu. On the **Image** tab, set the rotation angle of the image in clockwise degrees. This is equivalent to the `transform:rotate` property in CSS.

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 and on web pages at the position of the image while the image is loading and when the image is not found. On web pages, alternate texts are also used for accessibility.

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.

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 436. 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.

There are two types of tables: standard tables which are static in nature, and Dynamic Tables which have a variable number of rows depending on a detail table in the record; see "Dynamic table" on page 344.

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 221.

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 http://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. Use the **Location** drop-down to select where to insert the element:
   - **At cursor position**: The table is inserted where the cursor is located in the template.
   - **Before element**: The table is inserted before the current HTML element where the cursor is located. For example if the cursor is within a paragraph, insertion occurs before the <p> tag.*
   - **After start tag**: The table is inserted within the current HTML element, at the beginning, just after the start tag.*
   - **Before end tag**: The table is inserted within the current HTML element, at the end, just before the end tag.*
   - **After element**: The table is inserted after the current element where the cursor is located. For example if the cursor is within a paragraph, insertion occurs after the end tag of the paragraph (</p>).*

   * 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 until the root of the body.
4. Click **Next** and select which fields should show up in the Dynamic Table.
   The order of the fields indicates in which order columns are displayed in the dynamic table, from left to right. Select a line and then use the **Up** and **Down** buttons to change the order of the columns.
   You could change the placeholder for each data field as well; just click a placeholder to edit it.
5. Click **Next** and use the drop-down to select the desired table style.
6. Uncheck the box **Resizable columns** if the columns should not be resizable from the Design and Preview modes in the workspace. This is useful if the column size is determined in the Source mode or in a style sheet.
7. 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.

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.

Styling a Table

Tables can be styled using the **Format > Table** menu item, while individual selected cells can be styled using the **Format > Table Cell** menu item.

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 410.

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 (<div>), 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, copy-paste text into a template, or use the **Insert Lorem Ipsum** toolbar button to insert dummy text.

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 335.

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 PReS ConnectDesigner. Formatting options like bold, italic and formats like Heading 1, Heading 2 are maintained.

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 362.

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; see "Editing HTML" on page 221.

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 <p>. The paragraph should be followed by a closing tag: </p>.

A line break looks like this in HTML: <br>.

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 404.

In all templates you can use the fonts that are provided with the Designer, as well as imported fonts; see "Fonts" on page 416.

Capture OnTheGo

With the Designer you can create Capture OnTheGo templates. COTG templates are used to generate forms for the Capture OnTheGo mobile application. For more information about the application see this website: Capture OnTheGo.

COTG Forms

A Capture OnTheGo form is actually just a Web form, that you could add without a Form wizard, but the COTG Template Wizards include the appropriate JavaScript and styles to create user-friendly, responsive forms; see "Capture OnTheGo (COTG) templates" on page 433.

The **action** of the Capture OnTheGo Form element should specify a Workflow HTTP Server Input task that receives and handles the submitted data. The action will look similiar to this: **http://192.168.175.1:8080/actionname** (where **actionname** is the HTTP action of the HTTP Server Input task). For more information about specifying an **action**, see "Adding a Form" on page 266.

COTG Elements

There is a number of special Capture OnTheGo (COTG) elements. They can be used in a Web context within a Form element; for a description of all COTG elements, see "COTG Elements" on page 290.

Adding elements to a COTG Form

To add a COTG element to a Form or Fieldset, click inside the Form or Fieldset, select **Insert > COTG elements**, and choose the respective element on the menu. Now you can change the element's settings:

1. Add an **ID** (required) and, optionally, a **class**.

   > **Note**
   >
   > The ID will be copied to the `name` attribute of the element. The `name` attribute is what identifies the field to the receiving server-side script. To change the name, select the element after inserting it and type the new name on the **Attributes** pane.

   ID's and classes are also useful with regard to variable data (see "Personalizing Content" on page 322) and styling (see "Styling templates with CSS files" on page 396).
2. Type a label, or choose No label under Style, to omit the label. (For Label elements there are no other options to be set.)
3. If applicable, choose a style for the label (for the label of a Checkbox, for example, you can't set a style).
   - **Wrap input with label** places the input element inside the Label element.
   - **Attach label to input** ties the label to the input element using the `for` attribute of the Label element.
   - **Use label as placeholder** inserts the given label text in the placeholder attribute of the field.
   - **No style** omits the label altogether.

     > **Note**
     >
     > The first two label styles ensure that when the user clicks the label, the input element gets the focus.

> **Note**
>
> When adding an element to a Form or Fieldset, you cannot specify a `name`; the ID will be copied to the element's `name` attribute. After adding the element to the Form or Fieldset you can change the `name` on the Attributes pane. The `name` attribute of Form elements is sent to the server after the form has been submitted.

Note

When you add a COTG element to a template that you didn't start with a COTG template wizard, the Designer will automatically add the jQuery library and the JavaScript file cotg.js, so that the element works well. The Foundation style sheets will not be added. You only get those when you start creating a COTG template with a template wizard.

How to make COTG elements required

To make a COTG element required, or to change the validation of a COTG Form, right-click the element and choose **Validation settings**. Now you can change the Form's validation method and set the requirements and a message per field. For an explanation of the settings, see "Changing a Form's validation method" on page 269.

Testing a COTG template

For information about testing a COTG template, see this how-to: Testing a COTG template, and Send COTG Test.

Using JavaScript

JavaScript files, libraries and frameworks can be added to a template, to add widgets and other functionality to your Capture OnTheGo Forms; see "Using JavaScript" on page 447.

How to use COTG data in a template

When a user submits a Capture OnTheGo Form, the data are received by a Workflow HTTP Server Input task that receives and handles the submitted data. Even when no other tasks are present in that Workflow configuration, Workflow will output an XML file that contains the submitted data, in a location specified for the Send To Folder plugin in Workflow.
To be able to use that data in a template, you have to create a Data Mapping Configuration that extracts the data from the resulting XML file. Next, you can create a Designer template that uses this Data Mapping Configuration. Strings, base64-encoded strings and SVG data stored in data fields using the DataMapper can be used in a Designer template as they are.

Capture OnTheGo (COTG) templates

With the Designer you can create Capture OnTheGo templates. COTG templates are used to generate forms for the Capture OnTheGo mobile application. For more information about this application, see the website: Capture OnTheGo.

A Capture OnTheGo form is actually just a Web form, that you could add without a wizard, but the COTG Template Wizards include the appropriate JavaScript and styles to create user-friendly, responsive forms. They are built upon the Foundation framework.

Foundation

All Web Template Wizards make use of the Zurb **Foundation** front-end framework. A front-end framework is a collection of HTML, CSS, and JavaScript files to build upon. Foundation is a **responsive** framework: it uses CSS media queries and a mobile-first approach, so that websites built upon Foundation look good and function well on multiple devices including desktop and laptop computers, tablets, and mobile phones . Foundation is tested across many browsers and devices, and works back as far as IE9 and Android 2. See http://foundation.zurb.com/learn/about.html.

After creating a COTG template, the other contexts can be added, as well as other sections (see "Adding a context" on page 219 and "Adding a Web page" on page 443).

Creating a COTG template using a Wizard

To create a COTG template with a Template Wizard:

1. - In the **Welcome** screen that appears after startup and when you click the Home icon at the top right, choose **Browse Template Wizards**. Scroll down until you see the **Capture OnTheGo Starter** Template Wizards.
   - Alternatively, on the **File** menu, click **New**, expand the **Template** folder, and then expand the **Capture OnTheGo Starter** folder.
2. Select a template. There are 8 types of Web Template Wizards:
   - **Blank**. The Blank COTG Template has some basic design and the appropriate form, but no actual form or COTG elements.
   - **Bill of Lading**. The Bill of Lading Template is a transactional template that includes a detail table with a checkmark on each line, along with Signature and Date COTG elements. Use this wizard as a way to quickly start any new Zurb Foundation based form for Capture OnTheGo.
   - **Event Registration**. The Event Registration Template is a generic registration form asking for name, phone, email, etc.
   - **Event Feedback**. The Event Feedback Template is a questionnaire containing different questions used to rate an experience.
   - **Membership Application**. The Membership Application Template is a signed generic request form that can be used for memberships such as gyms, clubs, etc.

- **Patient Intake**. The Patient Intake Template is a generic medical questionnaire that could potentially be used as a base for insurance or clinic form.
- **Kitchen Sink**. The Kitchen Sink Template includes a wide range of basic form and COTG form elements demonstrating various possibilities of the software.
- **Time Sheet**. The Time Sheet Template is a single page application used to add time entries to a list. This template demonstrates the dynamic addition of lines within a COTG template, as the Add button creates a new time entry. There is no limit to the number of entries in a single page.

3. Click **Next** and make adjustments to the initial settings.
   - **Submit URL**: enter the URL where the form data should be sent. The URL should be a server-side script that can accept COTG Form data.
   - The **Title** and the **Logo** that you choose will be displayed at the top of the Form.
   - **Background color**: Enter a valid hexadecimal color code for the page background color (see w3school's color picker).
   - Enter a valid hexadecimal color code (see w3school's color picker) for the background color of the navigation bar at the top and another for the buttons on the Form.
4. Click **Next** to go to the next settings page if there is one.
5. Click **Finish** to create the template.

The Wizard creates:

- A **Web context** with one web page template (also called a **section**) in it. The web page contains an 'off-canvas' Div element, Header, a Section and a Footer element with dummy text, and depending on the type of web page, a navigation bar, button and/or Form elements.
- Style sheets and JavaScript files related to the COTG form itself and others related to the Foundation framework (see above). The style sheets can be found in the **Stylesheets** folder on the **Resources** pane. The JavaScript files are located in the **JavaScript** folder on the **Resources** pane.
- A collection of Snippets in the **Snippets** folder on the Resources pane. The Snippets contain ready-to-use parts to build the web page. Double-click to open them. See "Snippets" on page 393 for information about using Snippets.

The Wizard opens the Web section, so that you can fill the Capture OnTheGo form.

6. Make sure to set the action and method of the form. The **action** of a Capture OnTheGo form should specify the Workflow HTTP Server Input task that receives and handles the

submitted data. The action will look like this: **http://127.0.0.1:8080/action** (8080 is Workflow's default port number; 'action' should be replaced by the HTTP action of that particular HTTP Server Input task).
The **method** of a Capture OnTheGo form should be **POST** to ensure that it doesn't hit a data limit when submitting the form. The **GET** method adds the data to the URL, and the length of a URL is limited to 2048 characters.

Adding elements to a COTG template

To fill a Capture OnTheGo form, you can use special Capture OnTheGo Form elements, such as a Signature and a Barcode Scanner element. To insert an element, select **Insert > Capture OnTheGo Form elements**and click the respective element; see "COTG Elements" on the facing page. Web Form elements can also be used on COTG Forms (see "Forms" on page 266 and "Form Elements" on page 269) as well as text, images and other elements (see "Content elements" on page 220).

> **Tip**
>
> If the COTG Form replaces a paper form, it can be tempting to stick to the original layout. Although that may increase the recognizability, it is better to give priority to the user-friendliness of the form. Keep in mind that the COTG form will be used on a device and don't miss the chance to make it as user-friendly as possible.

Capture OnTheGo templates can be personalized just like any other type of template; see "Variable Data" on page 333 and "Personalizing Content" on page 322.

> **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.

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.

COTG Elements

With the Designer you can create Capture OnTheGo templates. COTG templates are used to generate forms for the Capture OnTheGo mobile application. This topic is about Capture OnTheGo form elements. For more information about the application see this website: Capture OnTheGo.

Capture OnTheGo (COTG) elements can only be added within a Form element in a Web context; see "COTG Forms" on page 284 and "Adding elements to a COTG Form" on page 284.

Barcode Scanner

The Barcode Scanner element adds a button to trigger the device to scan a barcode. A very large variety of barcode types are supported. Once the barcode has been scanned, the data from the barcode will be added to the COTG Form and submitted along with it.

Note

Using the Barcode Scanner element requires the installation of the ZXing Barcode Scanner application on Android devices. The application returns the barcode data after scanning.

Camera

The Camera element adds a group of buttons to capture or select an image. Once the image is selected via the camera or the device's library (aka "gallery"), it is saved within the Form data. When the form is submitted, the image is sent in a base64-encoded string format.

Buttons

By default, the Camera element adds three buttons to the form:

- **Take now**: Opens the device's default Camera application to take a picture using the device's camera. Capture OnTheGo has no impact on the device's applications, so the

features available (quality, orientation, filters) are dependent on the device itself. You can, however, set the format, quality and scaling for images that are submitted by the Camera element, as explained below.
- **Library**: Opens the device's default library or gallery application to select a single image that is then saved in the form data. The accessible images and navigation depend on the device itself.
- **Clear**: Removes any existing image data from the Camera element.

To omit the Take now or Library button, edit the Camera element's properties: right-click the Camera element after adding it to the form, select **Camera properties** and then use the **Source** drop-down to select which buttons will be available: Take, Pick from library, or both.

Annotations

Annotations can make a picture much more informative: an arrow, showing in which direction a car was driving; a circle, where the car has been damaged... To allow the user to make annotations, right-click the Camera element after adding it to the form, select **Camera properties**, and then check **Allow annotations**.
Clicking (or rather, touching) the image will bring up the annotation dialog. Annotations can be made in a Marker (semi-transparent) or Pencil (solid) style, in different colors and with different widths.
Annotations are submitted in SVG format by a hidden input added to the Camera element. The name of that input is the ID of the Camera element, followed by "-note-data", for example **camera1-note-data**.

Cropping/editing

To allow the user to crop or edit the image after taking or selecting it, select **Camera properties**, and then check **Edit Images**. Which editing options the user actually gets depends on the operating system of the device. On an Android device for example, the user may be able only to crop the image, while the user of an iOS device may select part of the image and rotate that selection.

Image format

You can set the format, quality and scaling for images that are submitted by the Camera element. Right-click the Camera element after adding it to the form, select **Camera properties** and edit the Image properties:

- **Format**: The image format can be PNG or JPG.
- **Quality**: Set the compression in a percentage.

- **Scale Image**: Check this option to enable image scaling. Then set the maximum width and height of images before they are sent to the server. Note that only the smallest of these is applied and the size ratio is always maintained.

How to use the captured or selected image in a template

When the user has submitted the form, you may want to use the captured or selected image in a Designer template, for example in a letter or on a web page. To do this, insert a dummy image in the template, right-click it and select **Dynamic Image**. The Text Script Wizard appears. Under **Field** select the field that contains the base64-encoded string. The script puts the given string in the source (**src**) attribute of the image (**<img>**).

Instead of using the Text Script Wizard, you could also write a script yourself; see "Write your own scripts" on page 373.

Date and Formatted Date

The Date element and the Formatted Date element display the current date on the device when the form is first opened. When the element is touched, a date selector appears so the user can modify this date. The Formatted Date element displays dates in a format that depends on the locale of the device on which the user is viewing the form. A Date Element displays dates in the ISO 8601 format: YYYY-MM-DD.

When the form is submitted, the date data is sent as plain text. A Formatted Date element submits the date in two formats: in the format that depends on the device's regional and language settings and in the ISO format mentioned above (using a hidden field). A Date element sends the date in the ISO format only.

Device Info

The Device Info Element adds a field that contains some information about the device (phone or tablet) that is submitting the COTG Form. This includes the device's type (Android or iOS), operating system version, device model and its UUID (Universally Unique Identifier). This information can be useful for both troubleshooting, if errors occur on specific device types for example, as well as for security validation: it is possible to maintain a list of device UUIDs that are allowed access, to prevent unauthorized use even if someone has a user name and password to a repository.

Fields Table

The Fields Table element adds a table with two rows, a delete button at the end of the first row and an add button at the end of the second row. Inside the rows you can put whatever elements you need. The user can click (or rather, touch) the Add button to add a row to the table. The new row will contain the same elements as the first row. The names of all elements in the first row will be extended with __0, while the names of the elements in the second row will be extended with __1, etc.

Geolocation

The Geolocation Element adds a button to read the device's current GPS coordinates and save them in a form field. When the button is pressed, the GPS coordinates are requested and saved. When the form is submitted, the Geolocation data is sent in plain text.

Image & Annotation

The Image & Annotation element is meant to be used with an image that needs input from the user. When inserting an Image & Annotation element you have to select the image. The user can simply click (or rather, touch) the image to bring up the annotation dialog. Annotations can be made in a Marker (semi-transparent) or Pencil (solid) style, in different colors and with different widths.
Annotations are submitted in SVG format by a hidden input. The name of that input is the ID of the Image & Annotation element, followed by "-note-data", for example **image_annotation1-note-data**.

Locale

The Locale Element does not have a UI element in the form. Inserting it adds a hidden input field that will contain the device's set locale when the form is submitted. This data is sent in plain text format and is available when processing the form data. The format is defined by the device.

Signature

The Signature Element adds a signature box to a COTG form. These signatures are filled in via touch input, either with a finger or capacitive pen. Touching the signature box opens up a fullscreen box used to sign (generally more useful in Landscape mode depending on the device); after confirming, the dialog saves the data into the Form.
Signature data is transmitted in SVG plain text format. To insert the signature data in a template, store the data in a field in a Data Mapping Configuration as is, and drag that field from the Data Model into a template to create a text script. The signature data can be inserted in the template's HTML directly, because the Designer (as well as web browsers) knows how to display this kind of data.

Time and Formatted Time

The Time element and the Formatted Time element display the current time on the device when the form is first opened. When the element is touched, a time selector appears so the user can modify this time. The Formatted Time element displays times in a format that depends on the locale of the device on which the user is viewing the form. A Time Element displays dates in the ISO 8601 format: HH:MM.

When the form is submitted, the time data is sent as plain text. A Formatted Time element submits the time in both the ISO format mentioned above and in the format that depends on the device's regional and language settings. A Time element sends the time in the ISO format only.

User Account

The User Account Element adds a hidden field which contains the Capture OnTheGo user account that was used to submit the form to the server. This is useful if a form is available to many different users, to detect who submitted it.

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 425. 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 219), the Email context folder is created along with other files that are specific to an Email context; see "Email context" on the next page.

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 297. 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 317.

Email templates are personalized just like any other template; see "Variable Data" on page 333.

Sending email

When the template is ready, you can change the email settings (see "Email header settings" on page 300) and send the email directly from the Designer or via Workflow. To test a template, you can send a test email first.

Email output can have the following attachments:

- The contents of the Print context, in the form of a PDF attachment
- The output of the Web context, as an integral HTML file
- Other files, an image, or a PDF leaflet for example.

See "Email attachments" on page 303 and "Generating Email output" on page 317.

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 425) or when an Email context is added to a template (see "Adding a context" on page 219) 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 297 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 297.
- 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 220 and "Email templates" on the next page.

Sending email

When the template is ready, you can generate Email output; See "Generating Email output" on page 317. To test a template, you can send a test email first. Email output can have the following attachments:

- The contents of the Print context, in the form of one PDF attachment.
- The default section of the Web context, as an integral HTML file.
- Other files.

> **Note**
>
> To split the Print context into multiple attachments, or to attach multiple Web sections as separate attachments, you need to create a Control Script that specifies *parts*; see "Control Script API" on page 208 and "Control Scripts" on page 385.

See "Email attachments" on page 303.

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, and right-click the **Email** context.
2. Click **Properties**.
3. 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 299.

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 220. 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.

Email templates are personalized just like any other template; see "Variable Data" on page 333.

The subject, recipients (To, CC and BCC), sender and reply-to address are specified with Email Script Wizards; see "Email header settings" on page 300.

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 317.

Adding an Email template

When an Email template is created (see "Creating an Email template with a Wizard" on page 425), 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 341), or Snippets and a script (see "Snippets" on page 393 and "Loading a snippet via a script" on page 383).

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 390).

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**.

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**
>
> 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. After closing and reopening the template it is no longer possible to restore the deleted context this way.

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 395.

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, the Designer converts all CSS rules that apply to the content of the email to inline style tags, as if local formatting was applied.

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 224.

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. Choose which CSS files should be applied to this section. You can also change the order in which the CSS files are read. This can have an effect on which CSS rule is applied in the end.

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**.

Note

The Default section is executed when the template is merged using the Create Email Content task in Workflow.

> **Tip**
>
> Use a Control Script to dynamically select an Email section for output depending on the value of a data field.

Email header settings

Email header settings define the information that goes into the header of each email that is generated from an Email template.

The default Email SMTP settings and the sender's name and address are defined in the preferences and can be adjusted in the Send Email and Send Test Email dialogs.

The subject, the recipients (To, CC and BCC), the sender and the reply-to address can be specified with Email Script Wizards.

Tip: Load data or a data mapping configuration first, so you can create Email Scripts that use a field in your data. See "Loading data" on page 324.

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 set defaults for when the output is generated (Production), for when you send a test email (Test) and for when you use Mandrill (Mandrillapp.com). The settings are:

- **Host**: The SMTP server through which the emails are to be sent. This can be a host (mail.domain.com) or an IP address. You can specify a port number as part of the host name, for example: smtp.mandrillapp.com:465.
- **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.)
- **Start TLS**: This option is enabled if authentication is checked. It sends emails through Transport Layer Security (TLS), which is sometimes referred to as SSL.

In the Send Email and Send Test Email dialogs you will be able to choose one of the three presets and adjust the settings to your needs.

Subject

To specify a subject for an email template:

1. On the **Resources** pane, expand the **Contexts** folder, and expand the **Email** section.
2. Right-click the section of which the subject should change and click **Properties**. Now you can change the subject.

Creating a dynamic subject with variable data

To replace an Email section's subject by a dynamic subject:

1. On the Scripts pane, click the black triangle on the **New** button and click **Email Subject Script**. A new script named Subject is added to the Scripts pane.
2. Double-click the new script to open it.
3. Select a data field and type a prefix and/or suffix.

The result of this script will appear in the email as the subject line.

To create a dynamic subject without variable data, or to create a subject that depends on the value of a data field, click **Expand** and modify the script. If you don't know how to write a script, see "Write your own scripts" on page 373.

Recipients: To, CC and BCC

The Email To Script Wizard defines the email address to which the email will be sent. The Email CC Script Wizard and the Email BCC Script Wizard define additional email addresses to which the email will be sent.

The To and CC email addresses will be visible to all of the recipients. The BCC ('blind carbon copy') email address(es) will not be visible to any of the other recipients.

To specify recipients for Email output:

1. On the Scripts pane, click the black triangle on the **New** button and click **Email To Script**, **Email CC Script** or **Email BCC script**. A new script is added to the Scripts pane.
2. Double-click the new script to open it.
3. Select a data field that holds an email address.

The result of this script goes in the email's To, CC, or BCC address field, respectively. It should be a valid, fully-formed email address.

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.

To dynamically specify a From address you have to use the Email From Script Wizard:

1. On the Scripts pane, click the black triangle on the **New** button and click **Email From Script**. A new script is added to the Scripts pane.
2. Double-click the new script to open it.
3. Select a data field that holds an email address.

The result of this script overwrites the address given in the Send Email dialog or Send Test Email dialog. It should be a valid, fully-formed email address.

Reply-To address

The Reply-To address is often used when sending email campaigns and to do tracking of email replies.

The Reply-To address has to be specified in a script:

1. On the Scripts pane, click the black triangle on the **New** button and click **Email From Script** or **Email Reply-To Script**. A new script is added to the Scripts pane.
2. Double-click the new script to open it.
3. Select a data field that holds an email address.

The result of this script should be a valid, fully-formed email address.

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. 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.

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.
- The output of the Web context, as an integral HTML file.
- Other files, an image or a PDF leaflet for example.

Attaching the Print context and/or the Web context is one of the options in the Send (Test) Email dialog; see "Generating Email output" on page 317.
By default, when adding the Print context to an email, all Print sections are output to a single PDF file, which is then attached to the email.

When adding the Web context to an email, only the default Web section is generated and added to the email as an HTML file.

> **Note**
>
> To split the Print context into multiple attachments, or to attach multiple Web sections as separate attachments, you need to create a Control Script that specifies *parts*; see "Control Script API" on page 208 and "Control Scripts" on page 385.

This topic explains how to attach files other than those generated by the Print or Web context. This is also described in a how-to; see Add custom email attachments.

Attaching external files

To attach files other than those generated by the Print or Web context to Email output:

1. Add the files to the template; see Adding images, or put them in a folder that is available to the machine that outputs the emails.

2. Create a script: on the **Scripts** pane at the bottom left, click **New**. A new script appears in the list. Double-click on it to open it. If you are not familiar with scripts, see "Write your own scripts" on page 373 for an explanation of how scripts work.
3. Change the name of the script, so that it reflects what the script does.
4. Choose the option **Selector** and in the **Selector** field, type **head**.
5. Write a script that appends a <link> element to the `results` (the selector is `head`, so the `results` contain the <head> of the email).
   - Make sure to set the **rel** attribute to **related**.
   - The **href** attribute determines where the file comes from. For resources inside of the template, use `'images/file.extension'` , or `'fonts/myfont.otf'`, etc. For external resources, you need the full path to the file, such as `'file:///c:/resources/attachments/instructions.pdf'`. Of course, you can also use dynamic calls such as `'file:///c:/clientfiles/' + record.fields.client_id + '/invoices/' + record.fields.invoice_number + '.pdf'`.

Examples

The following script attaches a PDF file named letter-CU00048376.pdf to each generated email. The PDF file is located in the Images folder on the Resources panel.

```
results.append("<link rel='related' href='images/letter-CU00048376.pdf'>");
```

If that same file would be located on the C: drive, the script should refer to it as follows: `href='file:///C:/letter-CU00048376.pdf'`.

The link doesn't have to be static; you could use data from the record set to build the link, for example:

```
var customerID = record.fields.ID;
results.append('<link rel="related" href="images/letter-' + customerID + '.pdf">');
```

Generating output

When merged with a record set, the templates made in the Designer can generate three types of output: Print, Email and Web.

Print output

Print templates, also called Print *sections*, are part of the Print context. They are meant to be printed to a printer or printer stream, or to a PDF file (see "Generating Print output" on page 306).

The Print context can also be added to Email output as a PDF attachment; see "Generating Email output"

on page 317. 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 385.

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 351.

To split the Print output into several files, see "Splitting printing into more than one file" on page 309.

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 318.

Email Output can be generated in two different ways: from the Designer or via Workflow. In both cases, 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.
- The output of the Web context, as an integral HTML file.
- Other files, an image or a PDF leaflet for example.

Attaching the Print context and/or the Web context is one of the options in the Send (Test) Email dialog; see "Generating Email output" on page 317.
To learn how to attach other files, see "Email attachments" on page 303.

Web output

The Web context outputs an HTML web page that contains the HTML text and all the resources necessary to display it.

Web output can be generated in two different ways: it can be attached to an Email template when generating Email output (see above), or it can be generated using Workflow; see "Generating Web output" on page 320.

Although the Web context can contain multiple Web pages, only one of them can be merged with each record. Which one is used, depends on a setting; see "Web output settings in the Web context and sections" on page 322.

Generating Print output

Print output can be generated from the Designer when a data set is available. The Designer merges all sections in the Print context with the database or data file, or with data from a Data Mapping Configuration.

> **Note**
>
> By default a Data Mapping Configuration does not contain the complete data set. Rather it contains a smaller subset of the data set, usually limited to 200 records. If you have an open Data Mapping Configuration, you can however open the complete data file to merge it with the Print template.

To generate Print output, select File from the menu and choose from any of these printing options:

- Select **File > Print...** to print using either of the following options:
  1. Using the **Default** output settings

     For more details, see "Print Using Standard Print Output Settings" on page 525
  2. Using the **last used** output settings

     For more details, see "Print Using Standard Print Output Settings" on page 525
  3. 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 behavior.

     For a detailed description see "Print Using Advanced Printer Wizard " on page 526.
  4. Using previously saved **Printing Preset** options.

     See "Job Creation Presets" on page 577 and Output Creation Presets for more details.

- Select **File > Proof Print...** to print using either the default output settings, the last used output settings or some previously saved output Presets.

  For more information on this option see "Print Using Standard Print Output Settings" on page 525. for more details.

Saving Printing options in Printing Presets.

Selecting **File > Print Presets** allows you to create or modify printing Presets, which can be saved and used in print runs thereafter.

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 "Job Creation Presets" on page 577 and "Output Creation Settings" on page 583 for more details.

Connect Printing options that cannot be changed from within the Printer Wizard.

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 influenced through either a job creation preset or output creation preset.

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. See "Enabling double-sided printing" on page 358.
- **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 352 and "Setting the binding style for a Print section" on page 358. Also see "Finishing Options" on page 578 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 360.

Adding print output types to the Print Wizard

Connect comes with several pre-prepared print output types. 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 PPML, VIPP and VPS).

To keep the Print Wizard manageable only a limited range of print output options are initially available in the Print Wizard. You can add other print output options to the selectable print options at any time, though.

How to add print output types from within the Print Wizard

Here is how to add print output options from within the Print Wizard dialog itself.

1. Select **File > Print...** from the menu. Print dialog is launched.
2. Click on the **Advanced** button. Print Wizard is launched.
3. Click the settings button ⚙ at the end of the Model selection.
4. Select **Edit available printers** from options.
5. In the Preferences dialog, select the print output options you would like to have added to the Print Wizard.

How to add print output types from within the Designer

Here is how to add print output options 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 section, select the print output options you would like to have added to the Print Wizard.

What's Next?

After you have added these printer options, the newly selected print output types will be available in the Print Wizard thereafter.

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. It is, however, possible to output one spool file per document, or to create groups of documents and store those in separate spool files.

Where the output should go, and how documents should be grouped, is set in a *Job Creation Preset*.

To make one document or a group of documents go into a separate file, the print job needs to be 'separated'. Separation is one of the options to set in an *Output Creation Preset*.

See "Generating Print output" on page 306 for a further explanation about Job Creation Presets and Output Creation Presets.

Variables available in the Output

In the Output Module, there are some variables available that offer more control over how templates are generated, or the data added to them.

Templates can be used in the following locations:

- The **Job Output Mask** field in the Print Options, when using the **Directory** option.
- The Text, Barcodes, OMR and Image data available in the Additional Content Options page.

Available Variables

The following is a list of variables that can be used independently of any job options loaded.

The Template object

${template}

Contains information about the template. The default use of `${template}` expands to a name based on the template name. A four digit sequence number is added at the end of the basename. The file extension is determined by the selected output technology. `${template}` is basically a short hand for `${template.base}_${template.nr,0000}.${template.ext}`

	The `0000` in `${template.nr,0000}` is a format pattern that takes care of formatting the number with at least four digits and leading zero's. See **Formatting date and number values**, below. **Example** If the template file is `C:\Data\My-Invoices-EN.OL-template` which gets printed to PDF, then `${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} this 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. 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**, below.
${template.ext}	The extension that corresponds to the chosen output technology. 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.

The File object

${file}	${file} is basically a short hand for ${file.base}_${file.nr,0000}.${file.ext} where 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**, below. **Server context:** On the Server, ${file} expands to a file name based on the job name. A four digit sequence number is added at the end of the basename. The suffix (the extension) is defined by the selected output technology. **Example** If the job name is my-invoices-reprint and is printed to PDF, then ${file} expands to my-invoices-reprint_0001.pdf **Designer context:** In the Designer, ${file} returns a generated name based on the current template name. A four digit sequence number is added at the end of the basename. The suffix is defined by the selected output technology. **Example** If the template file is `my-invoices.OL-template` and is printed to PostScript, then ${file} expands to `my-invoices_0001.ps`
${file.base}	The name of the template without dot extension (designer context) or the name of the job without dot extenstion (server context) **Example (Designer context)** If the template file is C:\Data\my-invoices.OL-template, then ${file.base} returns my-invoices
${file.ext}	The extension that corresponds to the chosen output technology. For example, for PDF output, ${file.ext} would be pdf, for PostScript output, ${file.ext} would return ps Note, that ${file.ext} does not include a leading dot.

${file.name}	The name of the template (designer context) or the name of the job (server context) **Example** If the template file is C:\Data\my-invoices.OL-template, then ${file.name} returns my-invoices.OL-template
${file.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**, below.
${file.pageCount}	The number of files in the current output file. It is possible to format the number using a pattern and locale. See **Formatting date and number values**, below.

The Job object

${job}	${job} expands to a name based on the job name. A four digit sequence number is added at the end of the basename. The file extension is determined by the selected output technology. ${job} is basically a short hand for ${job.base}_${job.nr,0000}.${job.ext} 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**, below. **Example** If the job name My-Invoices-Reprint.XY2015 gets printed to PDF, then ${job} expands to My-Invoices-Reprint_0001.pdf
${job.base}	Returns the base name of the job without any extension. **Example**

	If the job name is My-Invoices-Reprint.XY2015, then ${job.base} expands to My-Invoices-Reprint
${job.name}	Returns the name of the job. **Example** If the job name is My-Invoices-Reprint.XY2015, then ${job.name} expands to My-Invoices-Reprint.XY2015
${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**, below.
${job.ext}	The extension that corresponds to the chosen output technology. For example, for PDF output, ${job.ext} would be pdf, for PostScript output, ${job.ext} would return ps Note, that ${job.ext} does not include a leading dot.

Other available properties

These are various properties available to the Output module that are not part of a larger object:

${system.time}	Displays the current system data and/or time. Can be formatted using the **Formatting date and number values**, below.
${document.metadata.*propertyname*}	Value of a meta data property of the document. The *propertyname* must have been defined as a *Tag Name* on the *Document Tags* tab of the *Meta Data Options* page in the *AdvancedPrint Wizard.* Note: this is only available if *Separation* based on *Document* has been selected on the *Separation* page in the *AdvancedPrint Wizard*.
${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* page in the *AdvancedPrint Wizard*. Note: this is only available if *Separation* based on *Document Set* has

	been selected on the *Separation* page in the *AdvancedPrint Wizard*.
${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 Segment Tags* tab of the *Meta Data Options* page in the *AdvancedPrint Wizard*. Note: this is only available if *Separation* based on *Job Segment* or *Split At Exactly n Sheets* has been selected on the *Separation* page in the *AdvancedPrint Wizard*.
${job.metadata.*propertyname*}	Value of a meta data property of the job. The *propertyname* must have been defined as a *Tag Name* on the *Job Tags* tab of the *Meta Data Options* page in the *AdvancedPrint Wizard*. Note: this is only available if separation is disabled or if *Separation* based on *Job* has been selected on the *Separation* page in the *AdvancedPrint Wizard*.

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 locale	${system.time, "dd MMMM yyyy", nl}	4 juli 2009
${expression,,locale}	Apply a default format with the specified locale	${system.time,,nl}	4 juli 2009 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.

Generating Fax output

It is possible to generate Fax output from PReS Connect through the use of PDF/VT output. Here are the details on how to implement such a process.

Required Components

The following components are required in order to output to Fax:

- A PReS Image license which includes PReS Fax.
- A Job Preset adding the appropriate metadata fields
- An Output preset generating a PDF/VT file.
- A PReS Workflow process outputting to the PReS Fax task.

Job Preset Configuration

The following metadata fields must be added to the Metadata Options page:

- **FaxNumber**: The phone number where the fax will be sent. Often part of the data.
- **FaxInfo**: The description of the fax in both the PReSFax dialog box and the fax log file.

Output Preset Configuration

The following settings must be used in the Output Preset:

- In the Print Options, a PDF type should be selected, such as Generic PDF.
- In the PDF Options, the PDF Type should be set to PDF/VT

PReS Workflow Process

The following Workflow will produce Fax output:

- The four regular Connect tasks to generate print output:
  - **Execute Data Mapping**
  - **Create Print Content**
  - **Create Job** using the above *Job Preset*
  - **Create Output** using the above *Output Preset*. The task's **Output Management** must be set to be *Through Workflow*.
- The **PReS Fax** connector task set to Passthrough (the first "Document" on the list).

Generating Tags for Image Output

It is possible, even easy, to generate specific tags and indexes for PReS Image. This can be used to send email, archive with Search or output to image formats.

Required Components

The following components are required in order to output to Image:

- A PReS Imaging license.
- A Job Preset adding the appropriate metadata fields
- An Output preset generating a PDF/VT file.
- A PReS Workflow process outputting to the PReS Image task.

Job Preset Configuration

For email output, the following metadata fields must be added to the Metadata Options page:

- **ImageSendTo**: Add to have PReS Image use the specified field as the E-mail address to which to send the PDF file.
- **ImageSendCc**: Add to have PReS Image use the specified field as the E-mail address to which to send the PDF file as a carbon copy (CC).
- **ImageSendBcc**: Add to have PReS Image use the specified field as the E-mail address to which to send the PDF file as a blind carbon copy (BCC).
- **ImageSubject**: Add to have PReS Image use the specified field as the subject of the email that is sent.
- **ImageBody**: Add to have PReS Image use the specified field as the body of the email that is sent.

Note that the PDF file generated by the Print context is sent as an attachment to the email sent using the information above.

PReS Search Indexing

For PReS Search indexing, you can add your own custom fields. Each field that is not included in the above or in Generating Fax output is added as an index for PReS Search. For example you could add CustomerID and this would appear as the CustomerID index in Search. Yes, it's that easy!

Output Preset Configuration

The following settings must be used in the Output Preset:

- In the Print Options, a PDF type should be selected, such as Generic PDF.
- In the PDF Options, the PDF Type should be set to PDF/VT

PReS Workflow Process

The following Workflow will produce Image output:

- The four regular Connect tasks to generate print output:
  - **Execute Data Mapping**
  - **Create Print Content**
  - **Create Job** using the above *Job Preset*
  - **Create Output** using the above *Output Preset*. The task's **Output Management** must be set to be *Through Workflow*.
- The **PReS Image** connector task set to Passthrough (the first "Document" on the list). If sending Email, choose the "Send Email" option of the Output group. Otherwise, choose Archive Output, ensure the output type is PDF, and optionally fill in the PReS Search Database tab appropriately.

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 Output can be generated in two different ways: from the Designer or via Workflow. In both cases, 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.
- The output of the Web context, as an integral HTML file.
- Other files, an image or a PDF leaflet for example.

Attaching the Print context and/or the Web context is one of the options in the Send (Test) Email dialog; see "Generating Email output" above.
To learn how to attach other files, see "Email attachments" on page 303.

Before generating Email output

- 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).
- You may want to **rasterize** certain elements, such as <div> elements, business graphics, or headings with a special font type. Rasterizing converts the element to a JPG or PNG image. This is very useful to support as many clients as possible. For example, some

email clients may not support SVG, so converting a resource to JPG instead would ensure that most email clients would actually see the output.
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.

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 300. 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 subject of the email is a property of an email section. See "Subject" on page 301.
- The sender(s), recipient(s) and the subject can be set using Script Wizards; see "Email header settings" on page 300.
- Default SMTP settings can be set in the preferences; see "Email header settings" on page 300.
- 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 299.
  To dynamically select a section for output, use a Control Script; see "Control Scripts" on page 385.
- PDF attachments can be compressed to make the files smaller; see "Compressing PDF attachments" on page 296.

Generating Email output from Connect Designer

To generate Email output from the Designer:

1. Open a template with an Email context.
2. Load a data file or database compatible with this template. See "Loading data" on page 324.

Note that although data can be retrieved from an open Data Mapping Configuration, by default a Data Mapping Configuration does not contain a complete data set. Rather it contains a smaller subset of the data set, usually limited to 200 records. If you have an open Data Mapping Configuration, you can however open the complete data file to merge it with the template.

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 Email or Send Test Email for a description of all the options. Finally, click OK.

> **Note**
>
> **About testing emails**
>
> 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**.
>
> For a description of how to test your email for different email clients, see this how-to: Test your emails with Litmus. For more information on Litmus, please see http://litmus.com/

> **Tip**
>
> For a detailed description of how to use Mandrill with Connect to send and track emails, see the following how-to: Using Mandrill.

Generating Email output from Workflow

1. Open a template with an Email context.
2. Send the template to PReS Workflow; see "Sending files to Workflow" on page 422.
3. Create a process in PReS Workflow containing at least the following steps:
   - Any input that will capture a job file that is compatible with the data mapping configuration that is used.

- An Execute Data Mapping task to generate a valid record set.
- A Create Email Content task with the appropriate settings.

Generating Web output

The Web context outputs an HTML web page that contains the HTML text and all the resources necessary to display it.
JavaScript files are added to the <head> in the generated HTML file. They are useful to add special features such as those offered by jQuery and its plugins, or MooTools.
Stylesheets are also added to the <head> and are used just as they would be used in a regular web page.

Web output can be generated in two different ways: it can be attached to an Email template when generating Email output, or it can be generated using Workflow.

Web output can be generated from the Designer when a data set is available. The data can be retrieved from a database or data file, or from a Data Mapping Configuration.

Note that although data can be retrieved from an open Data Mapping Configuration, by default a Data Mapping Configuration does not contain a complete data set. Rather it contains a smaller subset of the data set, usually limited to 200 records. If you have an open Data Mapping Configuration, you can however open the complete data file to merge it with the template.

Before generating Web output

Before actually generating the Web output, you may want to **rasterize** certain elements, such as <div> elements, business graphics, or headings with a special font type. Rasterizing converts the element to a JPG or PNG image. This is very useful to support as many clients as possible. For example, when a heading uses a font type that is not a Web font, converting the heading to JPG instead would ensure that the heading looks the same in all browsers.
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.

Attaching Web output to an Email template

To attach the Web context to Email output:

1. Open a template with a Web context and an Email context.
2. Check the output settings for both contexts; see "Email output settings in the Email context and sections" on page 318 and "Web output settings in the Web context and sections" on page 322.

> **Note**
>
> When adding the Web context to an email, only the default Web section is generated and added to the email as an HTML file. To attach multiple Web sections as separate attachments, you need to create a Control Script that specifies **parts**; see "Control Scripts" on page 385 and "Control Script API" on page 208.

3. Load a data file or database compatible with this template. See "Loading data" on page 324.
4. On the **File** menu, click **Send Email** or **Send Test Email**. In the dialog that appears, check the option to attach the Web context to the email. See Send Email or Send Test Email for a description of all options.

> **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.

5. Fill in the dialog and send the emails.

Generating Web output from Workflow

1. Open a template with a Web context.
2. Send it to Workflow using the Package Files dialog; see "Sending files to Workflow" on page 422.
3. Create a process in Workflow containing at least the following steps:
   - Any input that will capture a job file that is compatible with the Data Mapping Configuration that is used. To capture incoming web requests, such as from a personalized URL (see "Personalized URL" on page 347), use a HTTP Server Input task.
   - An Execute Data Mapping task to generate a valid record set.
   - A Create Web Content task with the appropriate settings, to generate the HTML output.

Web output settings in the Web context and sections

There are a few settings for the Web context and Web sections that have an impact on the actual web page that is generated.

These settings are:

- Which Web section is the 'default'; see "Setting a default Web page for output" on page 445. When generating Web output, if there are multiple Web sections, only one of them can be merged with each record.
- The title, shortcut icon and meta tags appearing in the web page's header. See "Setting the title, meta data and a shortcut icon" on page 446

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.

Before you can start personalizing the content of a template, you must open a Data Mapping Configuration, data file or database; see: "Loading data" on page 324.

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 for each customer. Variable 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 web page, letter or email. Scripts in PReS Connect Designer are the basis of Variable Data Printing.

The easiest, quickest and most direct way to add customer data to content is via drag and drop; see "Variable Data" on page 333.

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 335.

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 341.

Dynamic images

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 343.

Dynamic tables

A Dynamic Table is a table with a variable number of rows that can overflow on one or more pages. It can also display subtotals on transport lines. In invoices, a Dynamic Table is an essential element. Read "Dynamic table" on page 344 to learn how to insert a dynamic table.

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 393 and "Loading a snippet via a script" on page 383.

Scripts

Self-made scripts

As soon as you want to do more than what can be done with the available (Text, Conditional) 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. For an introduction on this, see "Write your own scripts" on page 373.

Control Scripts

Control Scripts are scripts that affect the output of a template per record as a whole, instead of parts of the content. They are executed before the data is merged and can be used to control how different sections of the context are handled when the output is generated.

With a control script you can, among other things:

- Conditionally omit sections from print output
- Dynamically set the background image of a section
- Make the page numbering continue over all print sections
- 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
- Output one web page or another, based on the value of a data field

You need some knowledge of JavaScript to edit Control Scripts, just as for any other self-made scripts, because there is no Control Script Wizard; see "Write your own scripts" on page 373.

See "Control Scripts" on page 385.

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 have 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 Mapping Configuration, see "Loading a Data Mapping Configuration" on the next page
- a data file, see "Adding data from a data file" on page 326
- a database, see "Adding data from a database" on page 328.

A Data Model and sample data are part of a Data Mapping Configuration.

When you open a data file or a database, the Data Model will be derived from it **unless** there already is an open Data Mapping Configuration; in that case, the current Data Mapping Configuration will try to retrieve data from the file or database, using its own Data Model and extraction logic.

After opening a Data Mapping Configuration or 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.

Note

Although it is possible to load data from a data file or database in the Designer without creating a Data Mapping Configuration for it, generally the best way to extract data is by creating a Data Mapping Configuration. With a Data Mapping Configuration you can, among other things:

- Use the same data file with a different template, or use different kinds of data files with the same template.
- Load transactional or structured data. If there are detail lines, transactions, or any variable number of items to put into the template, you need to a Data Mapping Configuration to extract them.
- Format, transform, conditionally include/exclude and enhance data from the source file.
- Use Workflow to automate the extraction of data from this kind of data file.

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.

Loading a Data Mapping Configuration

If you have used the DataMapper first, you probably already have an open Data Mapping Configuration. Its Data Model and sample data will automatically be used when you start creating a template. You might have to click the **Synchronize Model** button on the **Data Model** pane, to update the fields.

To open a Data Mapping Configuration:

1. Open the Welcome screen: click the Home icon at the top right or select **Help > Welcome** on the menu.

2. Click **Open an existing configuration**.
3. Select the Data Mapping Configuration and open it.
4. At the top of the workspace, click the tab with the name of the template's section to go back to the template.
5. Click the button **Synchronize model** at the top of the **Data Model** pane to reload the data model.

> **Note**
>
> When generating output with just a Data Mapping Configuration, the template is merged with the complete sample data file that is part of the Data Mapping Configuration. The output is **not** limited to the number of records shown in the Data Model pane (which is one of the settings in the DataMapper).

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:
   - CSV files (.csv)
   - Microsoft Access Database (.mdb, .accddb)
   - XML files (.XML)
   - PDF/VT files
   - AFP files.
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).

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.

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.

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.

PDF/VT file options

After selecting a file, use the drop-down to select what level in the PDF/VT file defines a record in your data. The names of the levels are taken from the PDF/VT file itself. (See "About PDV/VT files" below.)

All metadata fields that belong to the chosen level and levels higher up in the tree structure will be listed. The lower the chosen level is in the tree structure, the more records you will get and the more metadata fields will appear in the list.

Select metadata fields to add them to your data. Their property names will be used as field names in the Designer's data model.

About PDV/VT files

The pages in PDF/VT files can be grouped on several levels. PDF/VT files can have a variable number of levels in their tree structure. The level's names are variable as well, with the exception of the lowest level, which is always called the **page** level. Metadata can be attached to each level in the structure.

AFP file options

After selecting a file, use the drop-down to select what level in the AFP file defines a record in your data. The levels are defined in the AFP file itself. (See "About AFP files" below.)
All metadata fields that belong to the chosen level and higher levels in the tree structure will be listed. The lower the chosen level is in the tree structure, the more records you will get and the more metadata fields will appear in the list.
Select metadata fields to add them to your data. Their property names will be used as field names in the Designer's data model.

About AFP files

Pages in AFP files are arranged in a tree structure, comprising one **document** at the top of the structure, **pages** at the bottom of the structure and one or more levels of **page groups** in between. (Unlike in PDF/VT files, the names of levels in AFP files can not be chosen freely.) In other words, an AFP file always consists of one document, that can contain page groups, of which each can be divided in page groups, and so on; the page groups at the lowest level contain pages.
Metadata can be attached to each level in the structure.

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:
   - 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.

MySQL

1. Enter the appropriate information to connect to the database:
   - **Server**: Enter the server address for the MySQL database.
   - **Port**: Enter the port to communicate with the 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 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.
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.
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 username 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 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.
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.
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.
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.
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.

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 for each customer. Variable 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 web page, letter or email. Scripts in PReS Connect Designer are the basis of Variable Data Printing.

After loading a Data Mapping Configuration or data from a data file or database (see "Loading data" on page 324), 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.

Note

Web templates are personalized just like any other template. There are a few extra possibilities, though; see "Using variable data on a Web page" on page 444.

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 **text script** appears in the **Scripts** pane at the bottom left.

A **text script** replaces 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 the text that the script will replace. The selector that the drag-and-drop method generates for a script, is the same as the placeholder that is placed in the text.

When you drag the same field 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.

> **Note**
>
> Looking for text in a text is a less optimized operation and may impact output speeds in longer documents. To speed up the output process, you could use a different type of selector. See "Using the Text Script Wizard" below for an explanation about the various types of selectors.
>
> **Tip**: press the **Alt** key while dragging, to wrap the placeholder in a span, give the span an **ID** and have that ID used as the script's selector.

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 so that it reflects what the script does.
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 see which parts of the template 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 HTML elements and/or CSS selectors. When speed matters, select one of the two remaining options: **Selector** or **Selector and Text**. See also "Testing scripts" on page 380.

- 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 395 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 http://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 <br/> (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. See "Formatting variable data" on the facing page.
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 262 and "How to position elements" on page 407).

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.

Formatting variable data

When a Text Script, made with the Text Script Wizard (see "Using the Text Script Wizard" on page 335) 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.

The available formatting functions depend on the data type of the corresponding field in the Data Model. In a Data Mapping Configuration you can set the data type of each field. When you open a data file or database without a Data Mapping Configuration, all fields are text fields (fields of the type `string`).

You could also format data in a script using the `formatter` ; see "API" on page 171.

Date

Dates in variable data can be displayed as long, medium and short dates with different time displays. There are quite a few presets, but you can also enter a custom format mask.

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 335.
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** you can choose one of the following options:
   - **Short Date** displays the day, month and year in two digits each, for example **01.04.16**.
   - **Medium Date** displays the day and month in two digits each and the year in four digits, for example **01.04.2016**. (This is also the value of the Default Date.)
   - **Long Date** displays the day as a number, the month's full name and the year in four digits, for example **1. April 2016**.
   - **Short Time** displays a time in hours and minutes in two digits each, for example **00:00**.
   - **Medium Time** displays a time in hours, minutes and seconds in two digits each, for example **00:00:00**. (This is also the value of the Default Time.)
   - **Long Time** displays a time in hours, minutes and seconds in two digits each, and adds a time zone, for example **00:00:00 EDT**.
   - **Short Date/Time** displays the date as a short date and the time as a short time, for example **01.04.16 00:00**.
   - **Medium Date/Time** displays the date as a medium date and the time as a medium time, for example **01.04.2016 00:00:00** (This is also the value of the Default Date/Time.)
   - **Long Date/Time** displays the date as a medium date and the time as a medium time, for example **1. April 2016 00:00:00 EDT**.

   Alternatively, you can enter a **custom format mask**: click in the **Format** column for the corresponding field and start typing a **pattern** to format the date (and optionally, the time). Do not put the pattern in quotes. For possible patterns see Date and time patterns.
   Note that this will only work if the type of the field has been set to Date in the Data Mapping Configuration and if the field contains a valid date.

   > **Note**
   >
   > The locale influences the way dates, times, numbers and currencies are formatted; see "Locale" on page 418.

4. Close the Script Wizard. For a new script, don’t forget to add the selector to the template.

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 335.
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. There are a few presets, but you can also type a format mask.

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 335.
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:
   - **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 418.
   - **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 418.
   - **Currency no symbol** does the same as Currency, but omits the currency symbol.
   - **Leading zero** adds a leading zero to a floating value between 0 and 1. This format is only available for fields that contain a `float` value. Note that when you open a data file or database without a Data Mapping Configuration, all fields are of the type `string`.
   - **∑ (Sum)** and **∑↑ (Sum Up)** are used in Dynamic Tables in a Print context. ∑ is for transport rules at the end of a page and ∑↑ shows the subtotal of the previous page.

   Alternatively, you can enter a **custom format mask**: click in the **Format** column for the corresponding field and start typing a **pattern**. 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 and http://docs.oracle.com/javase/7/docs/api/java/text/DecimalFormat.html. Note that for this to work, in the DataMapper the field that contains the value must be set to SmallInteger, BigInteger, Float, SmallCurrency or LargeCurrency.

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 a field's value. 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.

Use the Conditional Script Wizard to show or hide one element – a paragraph, image or other HTML element - based on the value of a data field.

Showing or hiding elements using the Conditional Script Wizard

1. Right-click the element 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 Content Script**. The Conditional Script Wizard opens.
2. Rename the script so that it reflects what the script does.
3. If you have started creating the script from the **Scripts** pane, you have to 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 335 for further explanation on selectors.
   If you have started the Conditional Script Wizard by right-clicking an element, you don't have to set a selector. If the element didn't have an ID, a new ID has been generated automatically. The new ID functions as the selector of the script.
   You can change the selector after closing and reopening the script (double-click the name of the script in the **Scripts** pane).
4. Set the **Action**: use the drop-down to select whether to **Show** or **Hide** the element when the condition below is true.
5. Click the downward pointing arrow next to **Field**, to select the data field that should be evaluated.
6. Click the downward pointing arrow next to **Condition** to expand the list of conditions with which the data field can be evaluated. The options are: **Equal to**, **Not equal to**, **Contains**, **Does not contain**, **Begins with**, **Ends with**.

7. Type the **Value** that should be used for the conditional check.
   For example, you could check whether the data field **Gender** is 'Equal To' the value 'M', in order to show a paragraph or an image applying to male customers only.
   If the condition evaluates to **true**, the selected action will be performed. If, conversely, the condition evaluates to **false**, and the option **Toggle Visibility** is checked, the opposite action will be performed. By default, this option is checked.

   > **Note**
   >
   > To combine the values of two or more data fields, you have to click **Expand** and edit the code of the script. See "Write your own scripts" on page 373.

8. Click **Apply** or **OK**.
9. To see the result, toggle to the **Preview** tab at the bottom of the workspace (or select **View > Preview View** on the menu).

Showing or hiding several elements with one conditional script

To apply one conditional content script to several elements, you have to use a CSS **class** or HTML element as the selector of the script. When using a CSS class, apply that class to the elements in question:

1. Double-click the conditional script in the **Scripts** pane to reopen it, or create a new conditional content script and follow the actions described in "Showing or hiding elements using the Conditional Script Wizard" on the previous page.
2. Change the selector to a CSS class (for example, .male) or to an HTML element with a certain CSS class (for example, p.male). See "Using the Text Script Wizard" on page 335 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 partially show or hide a paragraph or a line item in a list. Before you

can do that, you have to select the text that you want to be shown or hidden and wrap it 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 Script Wizard" on page 341.

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.

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 group of customers. All files should be of the same type and they need to be stored in one folder (the **Images** folder on the **Resources** pane, or an external folder). It is important that they are named after the various possible values of the related data field. Adding dynamic images that are not named after a data field value requires a self-made script.

To use the Dynamic Image Script Wizard:

1. Add one image to the template. See "Adding images" on page 277.
2. Right-click the image and click **Dynamic Image**. 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.
   The Dynamic Image Script Wizard composes a file name (including the path) based on the value of a data field, a prefix and a suffix:

- The prefix shows the path of the image.
- The suffix states the file extension of the image.
- The file name is the value of the data field(s) in the **Field** column.

The prefix and suffix are derived from the current image.

3. If necessary, enter another **Prefix** and/or **Suffix**.
4. Click the first field in the column **Field**, and then click the downward pointing arrow. Select the data field to be evaluated.
   Click the button **Add**, to add more fields if you want the file name to be composed of the value of several data fields. Note that only the suffix of the last data field should hold the file extension.
   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.
5. 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, as described in this how-to: 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.

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.

Dynamic table

In invoice templates, a Dynamic Table is an essential element. A Dynamic Table is different from a standard table in that it has a variable number of rows. In a Print context it can overflow on one or more pages and display a transport line.

Dynamic Tables are only available when the loaded record set or Data Mapping Configuration contains transactional data in one or more **detail tables** (see "Loading data" on page 324).

Creating a Dynamic Table

To create a Dynamic Table:

On the toolbar, click the **Insert detail table** button, or on the menu select **Insert > Table > Dynamic**.

1. 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.
    - **Detail Table**: Use the drop-down to select which detail table to display within the dynamic table.
    - **Width**: Enter the width of the table.
2. Use the **Location** drop-down to select where to insert the element:
    - **At cursor position**: The table is inserted where the cursor is located in the template.
    - **Before element**: The table is inserted before the current HTML element where the cursor is located. For example if the cursor is within a paragraph, insertion occurs before the <p> tag.*
    - **After start tag**: The table is inserted within the current HTML element, at the beginning, just after the start tag.*
    - **Before end tag**: The table is inserted within the current HTML element, at the end, just before the end tag.*
    - **After element**: The table is inserted after the current element where the cursor is located. For example if the cursor is within a paragraph, insertion occurs after the end tag of the paragraph (</p>).*

    * 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 (see "Selecting an element" on page 224), from the current selection point until the root of the body.
3. Click **Next** and select which fields should show up in the Dynamic Table.
    The order of the fields indicates in which order columns are displayed in the dynamic table, from left to right. Select a line and then use the **Up** and **Down** buttons to change the order of the columns.
    You could change the placeholder for each data field as well; just click a placeholder to edit it.

4. Click **Next** and check **Calculate Subtotals** to enable the options for a (sub)total at the end and (in Print sections) transport lines. The options are:
   - **Column**: Use the drop-down list to select the field that contains the currency value to be used to calculate the subtotal of the table. This field generally contains the result of item prices multiplied by the quantity.
   - **Field name**: Type the name to display in the footer when displaying that page's subtotal.
   - **Show in footer**: Check to display the subtotal in the footer of the table at the bottom of each page **and** at the end of the table.
   - **Show in header** (transportline): Check to display the subtotal of the previous page at the top of the table.
5. Click **Next** and select the styling attributes. Use the drop-down to select the desired table style.
6. Click **Finish** to add the table to the section.

When a Dynamic Table is added, a script is created for each of the columns containing a placeholder for a field that is to be replaced. These scripts are placed inside a folder named after the table's ID, on the **Scripts** pane.

Adding a row at the bottom of a Dynamic Table

Rows for extra information, such as the taxes or the total of the invoice, can be added at the bottom of a Dynamic Table as follows:

1. In the workspace, open the **Design** tab. Right-click the last line of the table and on the shortcut menu select **Row > Insert below**.
2. Drag a data field in the last cell of the new row (see "Variable Data" on page 333).
3. Select the row, either using the **breadcrumbs** (see "Selecting an element" on page 224) or using the toolbar: click the **Select Table** button and then click the **Select Row** button.
4. On the **Attributes** pane, change the **Show row** attribute to **At end of table**, to ensure that the row will only appear at the bottom of the Dynamic Table and not before each page break.
5. Add a label: click in the cell to the left of the data field and type the label.

If the subtotal at the end of the table is the actual total, you could just add a heading in the column to the left of the subtotal: click the table cell to the left of the subtotal and type the label. Note that in the Workspace the Design tab must be open for this to work. To open the Design tab, click it (at the bottom of the Workspace) or select **View > Design View**.

Creating a custom overflow footer

For an example of how to create a custom overflow footer, see the following how-to: Custom table overflow footers.

Change detail line formatting based upon a data field value

For an example of how to change the formatting of a line in a Dynamic Table, based upon a data field value, see the following how-to: Change detail line formatting based upon a data field value.

Personalized URL

Personalized URLs (pURLs) are links that are tailor-made for a specific purpose, generally for individual clients. They can serve multiple purposes, for instance:

- **Click Tracking**: A unique ID in the link makes it possible to track the source of the click (for example, a link in an email campaign).
- **User Tracking**: A user-specific ID reveals who clicked the link and at what time.
- **Landing Pages**: Information in the link invokes a unique landing page with specific products or services.
- **Personalized User Pages**: Using information from a database, a user is served a completely personalized web page with their name and information tailored to them, enhancing user response.

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.

In addition to the pURL, to generate a personalized landing page the Connect Server needs a template with a Web context and a Workflow process with the following tasks:

- A HTTP Server Input task to capture incoming web requests.
- An Execute Data Mapping task to create the record set appropriate for the template.
- A Create Web Content task that generates the HTML files.

Creating a personalized URL

Creating a personalized URL implies writing a script. See "Write your own scripts" on page 373.

It also requires some planning, because the pURL needs to contain data that is necessary to create the web page. For instance, creating a personalized URL for a client's invoice may require the Invoice Number to be present in the URL, which is then used to retrieve the invoice data, generate the invoice in PDF or HTML format using a template, and then return it to the browser. The trick is then to add the designated information to a hyperlink. How to do this is described in a how-to; see How to dynamically insert a hyperlink.

Print

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 etc.

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 to a printer or printer stream, or to a PDF file (see "Generating Print output" on page 306).
The Print context can also be added to Email output as a PDF attachment; see "Generating Email output" on page 317. 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 (see "Creating a Print template with a Wizard" on page 423), or when a Print context is added to an existing template (see "Adding a context" on page 219) the Print context folder is created along with other folders and files that are specific to a Print context; see "Print context" on the next page.

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 352.

Pages

Unlike emails and web pages, 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 360.

Although generally the same content elements can be used in all three contexts (see "Content elements" on page 220), the specific characteristics of pages make it possible to use special elements, such as page numbers; see "Page numbers" on page 362.

See "Pages" on page 359 for an overview of settings and elements that are specific for pages.

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 section 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 365 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 306.

See "Media" on page 368 for further explanation about how to add Media and how to apply them to different pages.

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 to a printer or printer stream, or to a PDF file (see "Generating Print output" on page 306).
The Print context can also be added to Email output as a PDF attachment; see "Generating Email output" on page 317. 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 423), or add the Print context to an existing template (see "Adding a context" on page 219).

> **Tip**
>
> To create a Print section from an existing PDF file, use a PDF file as a Print section's background. 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. See "Using a PDF file as background image" on page 356.

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 "Print context" on the previous page.
  See "Print sections" on page 352 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 section 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 365.
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 368 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 396.

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 the facing page.

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. This can be done using a Control Script; see "Control Scripts" on page 385.

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" on page 358. 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, and right-click the **Print** context.
2. Click **Properties**.
3. Choose a Binding style and, if applicable, the number of holes.

To set the binding style of a Print **section**, see "Setting the binding style for a Print section" on page 358.

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" on page 577 for more details.
2. Select that Job Creation Preset in the Print wizard; see "Generating Print output" on page 306.

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 360.

Print sections

Print templates, also called Print *sections*, are part of the Print context. They are meant to be printed to a printer or printer stream, or to a PDF file (see "Generating Print output" on page 306).
The Print context can also be added to Email output as a PDF attachment; see "Generating

Email output" on page 317. 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 and web pages, 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 360.

Although generally the same content elements can be used in all three contexts (see "Content elements" on page 220), the specific characteristics of pages make it possible to use special elements, such as page numbers; see "Page numbers" on page 362.

See "Pages" on page 359 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 section 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 365 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 368 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 306.

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 423 and "Print context" on page 349), 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**.

The first Master Page (see Master Page) and Media (see Media) 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 367 and "Applying Media to a page in a Print section" on page 371.

Tip

To create a Print section from an existing PDF file, use a PDF file as a Print section's background. 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. See "Using a PDF file as background image" on page 356.

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**
>
> 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. After closing and reopening the template it is no longer possible to restore the deleted context this way.

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 395.

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. Choose which CSS files should be applied to this section. You can also change the order in which the CSS files are read. This can have an effect on which CSS rule is applied in the end.

Using a PDF file as background image

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 or from a PDF file provided by the DataMapper. 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.

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 either **From Datamapper input** or **From PDF resource**.
   From Datamapper input uses the active Data Mapping Configuration to retrieve the PDF file that was used as input file, or another type of input file, converted to a PDF file. With this option you don't need to make any other settings; click OK to close the dialog.
3. For a PDF resource, you have to specify where it is located. Click the **Select Image** button.
   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** lets you choose an image file that resides in a folder on a hard drive that is accessible from your computer. Click the **Browse** button to select an image.
     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.
 Check the option **Save with template** to insert the image into the **Images** folder on the **Resources** pane.
- **Url** allows you to choose an image from a specific web address. Select the protocol (**http** or **https**), and then enter the web address (for example, http://www.mysite.com/images/image.jpg).

> **Note**
>
> It is not possible to use a remotely stored PDF file as a section's background, because the number of pages in a PDF file can not be determined via the http and http protocols. Therefor, with an external image, the option **Save with template** is always checked.

4. 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.
5. 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.

6. Finally, click **OK**.

Note

To set the background of a section in script, you need a Control Script; see "Sample scripts" on page 213.

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 352.

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.

To set the binding style of the Print **context**, see "Setting the binding style for the Print context" on page 352.

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" on page 577 for more details.
2. Select that Job Creation Preset in the Print wizard; see "Generating Print output" on page 306.

Enabling double-sided printing

To print a Print section on both sides of the paper, that Print section needs to have the Duplex printing option to be enabled.

Note

Your printer must support duplex for this option to work.

To enable duplex 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 **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" below to find out how to set a left and right margin on a page.
   - If an odd page count is generated, the last page (which is a duplex backside) has only the master page. To suppress the master page on this last page and exclude this page from page counting, check the option **Omit Master Page Back in case of an empty back page**.

Pages

Unlike emails and web pages, 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.

Although generally the same content elements can be used in all three contexts (see "Content elements" on page 220), the specific characteristics of pages make it possible to use special elements, such as page numbers; see "Page numbers" on page 362.

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 362.

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 365 and "Media" on page 368.

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 362 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" on page 344.

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 220.

When it comes to positioning elements on a page, Guides can be useful, as well as Tables. See "How to position elements" on page 407.

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 **Window** menu, click **Preferences**, click **Print**, and then click **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 322) 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 276 and "Snippets" on page 393.
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 224.
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
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 365.

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 367) 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.

By default, page numbering starts with page 1 for each section. This can be changed using a Control Script; see "Control Scripts" on page 385

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.

This setting can be changed for the entire Print context, per paragraph and in tables.

Note

Widows and orphans are ignored if the **page-break-inside** property of the paragraph is set to **avoid**.

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 number of lines that should be considered a widow or orphan (this amounts to the minimum number of lines that may be separated from a paragraph, minus one).

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. Select the paragraph, using the breadcrumbs or the **Outline** pane (next to the **Resources** pane).
2. On the **Format** menu, click **Paragraph**.
3. After **Widows** and **Orphans**, type the number of lines to be considered a widow or orphan (this amounts to the minimum number of lines that may be separated from a paragraph, minus one).

In tables

The CSS properties **widows** and **orphans** can be used in tables. They are not available in the Table Formatting dialog, however, so they must be added manually, either directly in the `style` attribute of the <table> element (on the **Source** tab in the Workspace) or in a style sheet rule, as follows:

1. On the menu, select **Edit > Stylesheets**.
2. Select the **Print** context.
3. Click **New** (or, when there are already CSS rules for tables, click the selector **table** and click **Edit**).
4. Click the **Advanced** button.
5. Add a rule for widows and/or orphans, typing the name of the CSS property in the left column and the value in the right column.
6. Close the dialogs.

Page breaks

A page break occurs automatically when the contents of a section don't fit on one page.

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 404):

- Select the element (see "Selecting an element" on page 224).
- On the **Format** menu select the respective element to open the Formatting dialog.
- 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.

> **Note**
>
> You cannot use these properties on an empty <div> or on absolutely 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 224).
- 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. This is equivalent to the **page-break-inside** property in CSS; see CSS page-break-inside property for an explanation of all available options.

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 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 section 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.

Master Pages resemble Print sections, and they are edited much the same way, see "Editing a Master Page" on the facing 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.

When a Print template is created, one master page is added to it automatically. You can add more Master Pages; see "Adding a Master Page" on the facing page. 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 367.

Examples

The following how-to demonstrates the use of Master Pages to show terms and conditions on the back of the first page of a Print section only: Showing a Terms and Conditions on the back of the first page only.

The following how-to shows how to use Master Pages to add a tear-off section to the first page of an invoice: A tear-off section on the first page of an invoice.

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 "Master Pages" on the previous 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 next page.

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.

A Master Page can contain text, images and other elements (see "Content elements" on page 220), including variable data and dynamic images (see "Personalizing Content" on page 322). It is good practice to position elements on a Master Page by placing them in a Positioned Box (see "Content elements" on page 220).

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" on the previous page.
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 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 a different Master Page and Media (see "Media" on the facing 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 and to skip that page from the page count.

5. Optionally, decide which Media should be linked to each sheet.
6. Click OK to save the settings and close the dialog.

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 371.

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 "Adding Media" 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.

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, import the PDF file by dragging it from the Windows Explorer to the **Images** folder on the **Resources** pane.
Then drag that the PDF file from the **Images** folder and drop it on one of the Media in the **Media** folder. With this method you can not 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 edit the properties of the Media.

Setting Media properties

Media have a number of properties that you can set, as described below. What you **cannot** set are a Media's page size and margins. The page size and margins are derived from the section to which the Media is applied.

You can, however, specify a PDF file (or any other type of image file) for both the front and the back of the Media, and specify 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 **Virtual Stationery**.
2. Click the **Select Image** button to select a PDF image file.
3. Click **Resources**, **Disk** or **Url**, depending on where the image is located.
    - **Resources** lists the PDF files that are present in the **Images** folder on the **Resources** pane.
    - **Disk** lets you choose an image file that resides in a folder on a hard drive that is accessible from your computer. Click the **Browse** button to select an image.
      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.
      Check the option **Save with template** to insert the image into the **Images** folder on the **Resources** pane.
    - **Url** allows you to choose an image from a specific web address. Select the protocol (**http** or **https**), and then enter the web address (for example, http://www.mysite.com/images/image.jpg).

> **Note**
>
> It is not possible to use a remotely stored PDF file as virtual stationery, because the number of pages in a PDF file can not be determined via the http and http protocols. Therefor, with an external image, the option **Save with template** is always checked.

4. Select a PDF file.
5. If the PDF file consists of more than one page, select the desired page.
6. Click **Finish**.
7. 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.
8. 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 365.

> **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.

Dynamically changing 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 "Write your own scripts" on the next page 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.

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 "Job Creation Presets" on page 577 for more details.
2. Select that job creation preset in the Print Wizard; see "Generating Print output" on page 306.

Write 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.

Most scripts can be made using one of the script Wizards. 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 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.

How scripts work

A script is a small set of instructions to the program, written in JavaScript.

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 (Control Scripts first).

All scripts, except Control Scripts, must have a **selector**. The selector can be text, an HTML element and/or a CSS selector. Running a 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 see which parts of the template 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.

Creating a new script

Writing a 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.
3. Choose which kind of **selector** you want to use. Running a 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, 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 them easier to recognize 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 without the angle brackets: p.
     - HTML elements with a specific CSS class (.green). In the script Wizard, click **Selector** and type the class name, including the preceding dot: .green.
     - An HTML element with a specific ID (#intro). In the script Wizard, click **Selector** and type the ID, including the preceding #: #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 http://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, 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 profiling scripts.

There is a shorter route to create a script for an element with a specific ID:

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. (In HTML, IDs start with #, but in this field you should type it without the preceding #).
3. Click the label to the left of the ID input field (ID)to make a new script with the ID that you typed as a selector.

Writing a script

1. Create a new script (see: "Creating a new script" on the previous page), 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: http://www.w3schools.com/js/js_syntax.asp and http://www.w3schools.com/js/js_if_else.asp.

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.

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**.
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** -

in this case they are cells with the CSS class 'amount' - 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.

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 "API" on page 171.

Managing scripts

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 facing page. What you can do is disable the script or the containing folder; see "Enable/disable scripts" on page 379.

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: "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 next 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 "Managing scripts" on page 377.
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.

Enable/disable scripts

A disabled script will not run at all when the template is merged with a record set to generate output.

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).

Import/export scripts

Scripts can be exported - one at a time - for use in other templates. To do this:

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 in a template:

- 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.

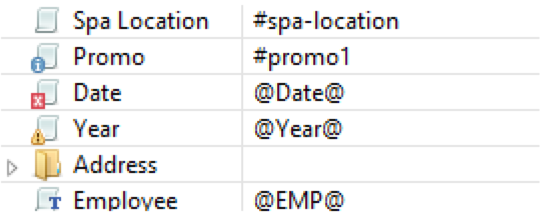
Note that scripts that use values of data fields can only be effective when a data file or Data Mapping Configurationis open. See "Loading data" on page 324.

Test for errors

Another way to 'test' a script is to take a look at the Scripts pane.

> **Tip**
>
> Hover over the name of a script in the **Scripts** pane to see which parts of the template are affected by the script.

Icons on the name of scripts in the **Scripts** pane can show a warning, information or error icon.



Spa Location	#spa-location
Promo	#promo1
Date	@Date@
Year	@Year@
Address	
Employee	@EMP@

- The information icon ⓘ (i) shows that the selector of the script does not produce a result in the current section.

- 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.

In addition to the icons and messages in the Scripts pane, there is another way to see if your scripts function as expected before generating output:

- On the **Context** menu, click **Preflight**.

Preflight executes the template without actually producing output and it displays any issues once it's done.

It will tell, for example, which selectors were not encountered in the template.

Test 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, 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 HTML and CSS selectors instead of text selectors to make the query faster.

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.

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 **Window** menu, click **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.

Loading a snippet via a script

Instead of dragging it into the content directly, it is possible, and often very useful, to load a snippet dynamically. Create a script (see "Write your own scripts" on page 373) and in the code, use the following function:

```
results.loadhtml(‘snippets/nameofthesnippet.html’)
```

This function will insert the snippet in the content at any position where the script's selector is encountered.

For more examples, see Skin/Formats/CrossReferencePrintFormat("loadhtml() Global function that replaces the content (inner html) of each matched element in the result set, alternatively load the data into a variable. The location should be an URL or a relative file path. loadhtml (location)loadhtml(location, selector) Loadhtml() is cached per batch run (based on the URL) in print/email. loadhtml(location) Loads all HTML from the 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 a HTML file residing in the Snippets folder on the Resources panel.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 variableThe 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('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); This last script loads a snippet into a variable and retrieves an element from the snippet using query().var mysnippet = loadhtml('snippets/text-root-wrapped.html'); var subject = query("#subject", mysnippet).text(); results.append("<p style='font-weight: bold;'>" + subject + "</p>");loadhtml(location, selector) Retrieve specific content from the filename. location String; the location can be absolute or relative to the section/context. Use: snippets/<snippet-name> to retrieve the content from a HTML file residing in snippets folder of the Resources panel.selectorString. The supplied selector should conform to CSS selector syntax and allows you to retrieve only the content of matching elements.ExamplesThis script loads a specific element from the snippet.var mysnippet = loadhtml('snippets/snippet-selectors.html','#item3'); results.replaceWith (mysnippet);This script loads the children of the selected element.var snippet = loadhtml ('snippets/snippet.html','foobar').children(); results.replaceWith(snippet);Another example is given in the following how-to: Using a selector to load part of a snippet." on page 1).

Note

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.

Loading part of a 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 the Designer API 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, <p ID=”En”>), 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: "API" on page 171.

Tip

An easy way to group content in a snippet is putting each part in a container and giving that container an ID, for example:
<div ID=”EN”><p>This is text for English customers.</p></div>
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 (<div>).

Load a snippet and insert variable data into it

The following script loads part of a snippet based on the value of a field, and then finds/replaces text by 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);
```

Control Scripts

Control Scripts are scripts that affect the output of a template per record as a whole, instead of parts of the content. They are executed before the data is merged and can be used to control how different sections of the context are handled when the output is generated.

With a control script you can, among other things:

- Conditionally omit sections from print output
- Dynamically set the background image of a section

- Make the page numbering continue over all print sections
- 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
- Output one web page or another, based on the value of a data field

You need some knowledge of JavaScript to edit Control Scripts, just as for any other self-made scripts, because there is no Control Script Wizard; see "Write your own scripts" on page 373.

This topic explains how to add a Control Script, how to use a Control Script to change the page numbering in Print output, and how to write a Control Script that splits one generated document into multiple files or attachments.

For examples and for more uses of Control Scripts, see the sample scripts in the Control Script API: "Sample scripts" on page 213.

> **Tip**
>
> New Control Scripts added to the template contain a few examples. For more examples see "Sample scripts" on page 213.

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 **New 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.
4. Write the script; see the "Control Script API" on page 208. If you are not familiar with scripting, also see "Write your own scripts" on page 373.
   Note: a newly added Control Script contains code to continue page numbering over all print sections, and two examples: one to select different sections of a Print context for an email and for print output, and one to select a web section.

Page numbering

Page numbering starts with page 1 for each section by default. If for a section **restartPageNumbering** is set to **false** (see "Control Script API" on page 208), that section will start with the page number following the last page of the previous section.

Assume that a template has four sections 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 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

Note that even if a section is not enabled, its **restartPageNumber** flag is still 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

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)

This allows a first section in a page sequence range to be optional without further scripting.

To mimic behavior of earlier software versions each section has `restartPageNumber = true` by default when the first control script runs.

Parts: splitting one generated document into multiple files

In a Control Script, *parts* can be defined to determine which sections should be output to the same file. This way it is possible, for example, to split the Print context or the Web context into multiple email attachments.

Defining parts is done by setting the **part** field on a *section*; also see "Control Script API" on page 208.

- If a part name is given, then that delimits the start of a new part.
- A part ends at the last enabled section or at the last section before the start of a new part.

If no part name is set on any section, it is assumed that there is only one part, consisting of the default section (for Web and Email output) or of all sections (for Print output).

Example 1: 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.

Example 2: 2 parts

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.

> **Note**
>
> For Web sections, a part always consists of only the given section. Web pages cannot be appended to form a single part. It is however possible to attach multiple Web pages to one email; see "Sample scripts" on page 213 for an example.

Print section background

In the Print context, a PDF file can be used as a section's background (see "Using a PDF file as background image" on page 356).

In a Control Script, you can do the same, and more: you could for example specify a particular PDF file as a section's background, depending on the value of a field in the current record.

The Control Script should first In case enable a background, in case an initial background wasn't set via the user interface. Enabling a background is done by setting the source type for the background of the section to either a DataMapper PDF or an arbitrary PDF. For an arbitry PDF, the Control Script should specify a path.
After that, the background can be positioned, setting the section's `background.position` to `ABSOLUTE` or to `FIT_TO_MEDIA`.

For examples, see the Control Script API: "Sample scripts" on page 213.

Sections

Sections are parts of one of the contexts in a template: Print, Email or Web. 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, Print or Web) on the **Resources** pane, and then click **New section**.

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

> **Tip**
>
> If an Email context is going to be part of the template, it is recommended to start with an Email Template Wizard; see "Creating an Email template with a Wizard" on page 425. After creating a template, contexts can be added to it, but that can not be done with a wizard.

Editing a section

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

Each section can contain text, images and many other elements (see "Content elements" on page 220), including variable data and other dynamic elements (see "Personalizing Content" on page 322).

Copying a section

Copying a section, either within the same template or from another template, can only be done manually. You have to copy the source of the HTML file:

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).
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.

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. 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**.

> **Note**
>
> Sections cannot have an integer as name. The name should always include alphanumeric characters.

Section properties

Which properties apply to a section, depends on the context it is part of. See also: "Print sections" on page 352, "Email templates" on page 297, and "Web pages" on page 442.

To change the properties for 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 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. Choose which CSS files should be applied to this section. You can also change the order in which the CSS files are read. This can have an effect on which CSS rule is applied in the end.

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 352, "Email templates" on page 297 and "Web pages" on page 442.

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 306.

In email and web output, only one section can be executed at a time. The section that will be output is the section that has been set as the 'default'. See "Generating Web output" on page 320 and "Web pages" on page 442 and "Generating Email output" on page 317 and "Email templates" on page 297. The 'default' section is always executed when the template is run using the Create Email Content task in Workflow.

It is, however, possible to include or exclude sections when the output is generated, or to set another section as the 'default', depending on a value in the data. A Control Script can do this; see "Control Scripts" on page 385.

See Skin/Formats/CrossReferencePrintFormat("Generating outputWhen merged with a record set, the templates made in the Designer can generate three types of output: Print, Email and Web.Print outputPrint templates, also called Print sections, are part of the Print context. They are meant to be printed to a printer or printer stream, or to a PDF file (see Skin/Formats/CrossReferencePrintFormat("Generating Print output" on page 1)). The Print context can also be added to Email output as a PDF attachment; see Skin/Formats/CrossReferencePrintFormat("Generating Email output" on page 1). 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 Skin/Formats/CrossReferencePrintFormat("Control Scripts" on page 1).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 Skin/Formats/CrossReferencePrintFormat("Print settings in the Print context and sections" on page 1).To split the Print output into several files, see Skin/Formats/CrossReferencePrintFormat("Splitting printing into more than one file" on page 1).Email outputThe 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 Skin/Formats/CrossReferencePrintFormat("Email output settings in the Email context and sections" on page 1).Email Output can be generated in two different ways: from the Designer or via Workflow. In both cases, 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. The output of the Web context, as an integral HTML file.Other files, an image or a PDF leaflet for example.Attaching the Print context and/or the Web context is one of the options in the Send (Test) Email dialog; see Skin/Formats/CrossReferencePrintFormat("Generating Email output" on page 1). To learn how to attach other files, see Skin/Formats/CrossReferencePrintFormat("Email attachments" on page 1).Web output The Web context outputs an HTML web page that contains the HTML text and all the resources necessary to display it.Web output can be generated in two different ways: it can be attached to an Email template when generating Email output (see above), or it can be generated using Workflow; see Skin/Formats/CrossReferencePrintFormat("Generating Web output" on page 1).Although the Web context can contain multiple Web pages, only one of them can be merged with each record. Which one is used, depends on a setting; see Skin/Formats/CrossReferencePrintFormat("Web output settings in the Web context and sections" on page 1)." on page 1) to learn how to generate Print documents, Web pages or Email.

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 and sections. They can contain any contents that a section can have, such as text, images, variable data, dynamic tables, etc.

When a snippet is added to a section or a master page, a reference to it is placed in the source code, but the content of the snippet itself is not added to the page. Modifying a snippet in a section that it has been added to, actually modifies the snippet's source. If a snippet is used in multiple locations (such as different contexts and sections), modifying one instance will modify all of them at once.

When a snippet is added to different sections or contexts, it is displayed according to the section's or context's stylesheet. This means that the same content can appear differently depending on the styles applied to the section or context, but it still has the exact same contents.

Adding a snippet

Before a snippet can be added to the content, the resource files that are related to the snippet, such as image files and CSS files, have to be imported 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 276.

Drag the snippet itself to the **Snippets** folder on the **Resources** pane, or create a new snippet from an existing piece of content in the template (see "Creating a snippet" below).

> **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.

To add the 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.

It is also possible, and often useful, to insert a snippet or part of it, using a script. See "Loading a snippet via a script" on page 383.

Creating a snippet

To turn a parts of a letter, email or web page into a snippet for reuse in the content of a template:

1. Select the part that should be saved in a snippet.
2. Right-click the selection, point to **Snippet** and click **Create**.
3. Right-click the new snippet on the **Resources** pane in the **Snippets** folder and rename it.

JSON Snippets

JSON Snippets are snippets that contain pieces of JSON data instead of HTML. Just like HTML snippets, JSON snippets are stored in the **Snippets** folder on the **Resources** pane, but their file names should end in '.json'.

JSON Snippets cannot be inserted into the content directly, but they can be accessed via a script using the function **loadjson()**:

```
var json_data = loadjson("snippets/snippet.json");
results.html(json_data.field1);
```

See also: "Write your own scripts" on page 373.

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 recommended to use style sheets in your templates right from the start, even more so if your communications are going to be output to different output channels, or if they consist out of different sections (for example, a covering letter and a policy). With CSS you can give your templates one look and feel. 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, for example. 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 413 and "Fonts" on page 416.

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 404.

Boxes and a number of other elements can have a background color and/or background image; see "Background color and/or image" on page 409.

Several elements, such as boxes, images, paragraphs, and tables, can have a border; see "Border" on page 410.

Boxes, images, tables, text and other elements can be rotated; see Rotating elements.

Spacing (padding and margin) helps to position elements relative to other elements in the template; see "Spacing" on page 419.

The best way to position elements depends on the output channel for which the template is intended; see "How to position elements" on page 407.

The locale setting influences how dates, numbers and amounts of money are displayed; see "Locale" on page 418.

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: Cascading Style Sheets.

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. 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.

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, Web). 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.

To create a new CSS file, right-click the **Stylesheet** folder on the **Resources** pane and select **New Stylesheet**.

> **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.

> **Note**
>
> The order in which style sheets are executed, can affect the actual output. This sequence can be set per section; see "Determining the order in which style sheets are read" on page 404.

Using a remote style sheet

A remote style sheet is not located within your template but is rather hosted on an external web server (generally called a **CDN**). When generating Web output, these files are referenced in the web page's header and are served by the remote server, not by the PReS Connect Server module.

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. This must be a full URL, including the http:// or https:// prefix, domain name, path and filename.

There are a few advantages to remote resources:

- 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.

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, the Designer converts all CSS rules that apply to the content of the email to inline style tags, as if local formatting was applied.

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.

- 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 http://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 395. Note: where a width can be set as a percentage, it is a percentage of the space between the margins.
6. 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.

Edit 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: http://www.w3schools.com/cssref/.

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.

> **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; t hey will both be affected by the CSS rules for the element with that ID.

Adding a class or ID to an HTML element

1. Select the element (see "Selecting an element" on page 224).
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”>`.

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 224) and take a look at the Styles pane that sits next to the Attributes 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 395).

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 the facing page 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 "Determining the order in which style sheets are read" on page 404.

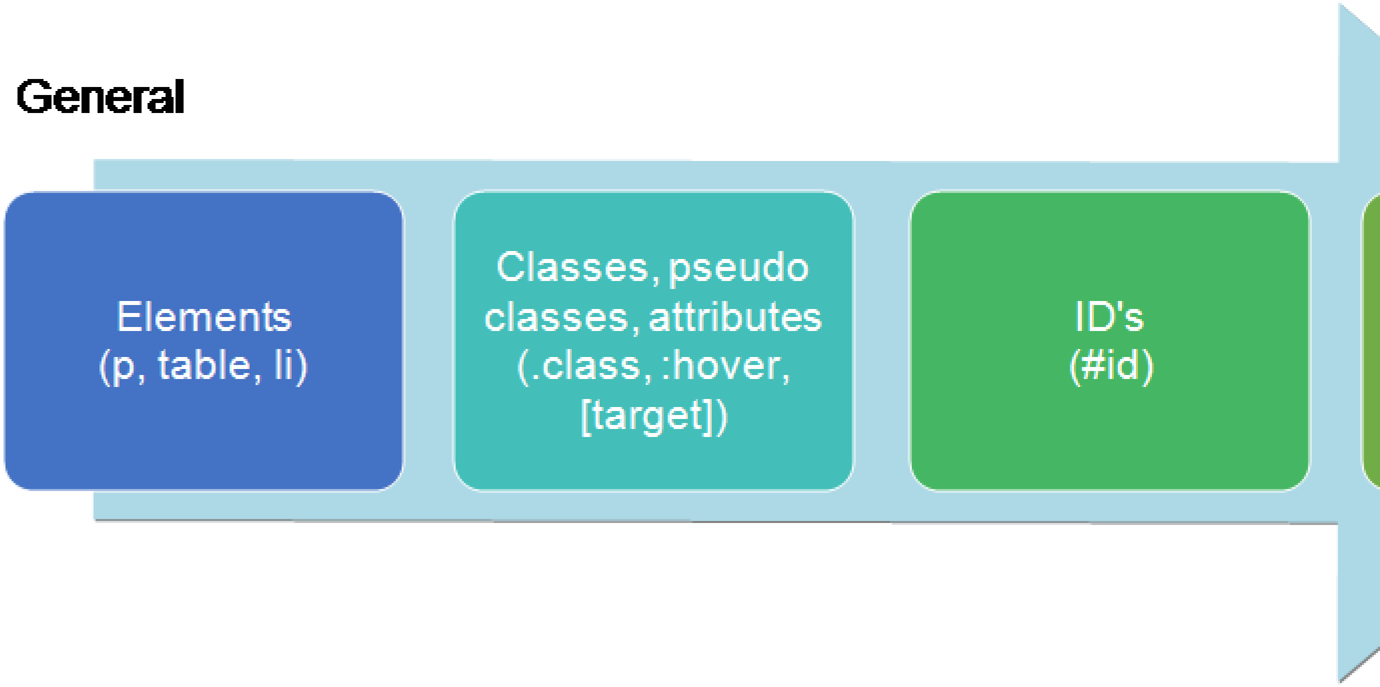
Using a more specific CSS rule

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 is also applied 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.


General
Elements
(p, table, li)
Classes, pseudo classes, attributes
(.class, :hover, [target])
ID's
(#id)

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

A more specific rule for cells in a table that has the CSS property “color: red” (which colors text in the cells red) 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.

Determining the order in which style sheets are read

For each section, the style sheets are applied in a certain order. 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.

The order in which style sheets are applied, can be changed per section:

1. On the **Resources** pane, expand the **Contexts** folder, expand the folder of the corresponding context and then right-click the template.
2. Click the tab **Includes**.
3. Click a CSS file and use the **Up** and **Down** buttons to change the order in which the style sheets are read.
4. Note: Moving a style sheet up in the list gives it **less** weight, because style sheets read later will override previous ones in case of conflicting rules.

Styling text and paragraphs

There are numerous ways to format text in a template. You can apply a certain font, make it bold, center the text, 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 395 and "Styling templates with CSS files" on page 396.

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 224) and click one of the toolbar buttons to make it bold, center it, create a numbered or bulleted list, etc.

To quickly change a paragraph into a Heading, Address or Pre element, select the paragraph (see: "Selecting an element" on page 224) and on the **Format** menu, select the appropriate element.

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**: see also: "Fonts" on page 416. 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**: this the color of the text. Select a named font color as defined in the Edit Colors dialog (see "Colors" on page 413) 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 setting is equivalent to the `color ` property in CSS.
  - **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 413) or click the colored square to create a new color or to enter a color value manually. a valid HTML color name or hexadecimal color code. 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
  - **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 395.

Formatting a paragraph

Through the Paragraph Formatting dialog you can set the line height and first indent, among other things. It also lets you add spacing and a border; see "Spacing" on page 419 and "Border" on page 410.

To open the Paragraph Formatting dialog, select a paragraph (see: "Selecting an element" on page 224) or place the cursor in a paragraph, and then select **Format > Text**.

On the **Formats** tab you can set:

- **Line-height**: Specify the height of each line in the paragraph's text, in a measure or percentage. Note that this is not the spacing between lines, but rather the complete height of the line itself including the text. This is 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 the paragraph. 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 in CSS.
- **Direction**: Select in which direction text should be displayed (left to right, right to left, or auto). Useful for certain languages such as Arabic, Hebrew, etc. This is equivalent to the `dir` HTML attribute.
- **(Page) breaks**: these settings are only useful in Print sections, as only Print sections have pages.
  - **Before**: Sets 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**: Sets 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**: Sets 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** and **orphans**: this setting is only useful in Print sections; see "Preventing widows and orphans" on page 362 for an explanation.

Click the button **Advanced** to add CSS properties and values to the inline style tag directly.

Remove 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: **<b>...</b>** or like this: **<p style= "color: red;" >**.

Inline style tags have priority over styles defined in a CSS file. 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 can 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.

How to position elements

To position elements in relation to each other in a template, wrap those elements in a Table or Box (see "Table" on page 280 and "Boxes" on page 262) 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 419. Guides help to align elements as well; see below.

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 436 and "Table" on page 280.

In the Web context, Inline Boxes are the preferred way to position elements; see "Boxes" on page 262. 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 (and often, Positioned Boxes) make those pages less accessible to users with disabilities and to viewers using smaller devices.

In the Print context, Tables can be used to position elements, as well as both types of Boxes; see "Table" on page 280 and "Boxes" on page 262.

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 419.

> **Tip**
>
> Use a negative left margin to create a hanging paragraph or image.

Guides

Guides are horizontal and vertical lines used to help in designing templates. Positioned Boxes (and any other objects that have their **position** set to **absolute**) will snap to guides when moved within a few pixels of them.

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. Double-clicking the guide brings up its Edit dialog where its exact position can be adjusted.

To delete a guide, double-click on it and press the **Delete** button.

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 224; note that a Box is a <div> element) and on the **Format** menu click **Box**.
3. Click the **Background** tab.

To select a **background color**: 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 413), or click the colored rectangle to open the Color Picker dialog; see "Dialogs" on page 451. In this dialog you can select a color from the color wheel, set RGB or CMYK color values or enter a hexadecimal color code.

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:http://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 396. 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 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 224) 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): 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 413), or click the colored rectangle to open the Color Picker dialog. In this dialog you can select a color from the color wheel, set RGB or CMYK color values or enter a hexadecimal color code. 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 396.

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 224) 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 http://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 224) 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 396. 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 395).

Defining colors

To do this:

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. This name should not contain spaces or special characters.

> **Tip**
>
> Working with style sheets? Choose a name that informs about 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 broad array of colors. This model is typically used for electronic devices.
   If applicable, check **Spot color**. Note that spot colors can only be used on certain printers.
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). If your printer can use spot colors and you want a spot color to be used in a Print context, you can define the color as described above, with two differences:

- Check the option **Spot color** instead of **Color**.
- Make sure that the color’s name matches that of the spot color used in the printer.

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. Click the Type drop-down and 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 t 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 recommended to use style sheets in your templates right from the start, even more so if your communications are going to be output to different output channels, or if they consist out of different sections (for example, a covering letter and a policy). With CSS you can give your templates one look and feel. 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, for example. See "Styling templates with CSS files" on page 396.

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 413), 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 224).
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 224).
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 413). 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 413; for an explanation of the other options in this dialog, see "Dialogs" on page 451. 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 452.

Fonts

In templates for personalized customer communications you can use the fonts that are provided with the Designer, as well as imported 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 224) 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 Styling with CSS.
  Note: The reason for specifying more than one fonts in a style sheet for web pages is that the font might not be available on the device on which the web page is viewed. Order the font names by preference. The last one should be 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.

When text is displayed in an imported font, the Designer can mimic the bold and italic versions of that font. If you have separate files for the bold, italic and possibly other versions of a font, you can make the Designer use the appropriate files to style text. To do this:

1. Import the files for the bold, italic and/other versions of the font into the Fonts folder.
2. On the **Edit** menu, click **Fonts**, to open the Font Manager.
3. Select the normal version of the imported font and duplicate it using the **Duplicate** button, once for each version of the font.
4. For each of the duplicates, combine a font effect with a file:
   - Click a duplicate and click the button Edit. Note: don’t change the duplicate’s name!
   - Select the appropriate font effect (font-weight and/or font-style).
   - Check the file or files the Designer should use for that effect. Per file type, one file can be checked.
5. Close the Font Manager.

The Designer currently supports 4 font types: TTF, OTF, WOFF, EOT and SVG.

When you are creating a Web template, keep in mind that the different font types are not supported by all clients; for instance, EOT and SVG are used only by Explorer and Safari, respectively.

If you're 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;
}
```

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 335 and "Formatting variable data" on page 338). 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 "API" on page 171.

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. This field must be a string and contain 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 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 as defined by ISO 3166.

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 224) 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 395.
>
> It is also possible to set an element's border in a style sheet; see Styling templates with CSS files.

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.

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 **Browse Template Wizards**. Scroll down to see all the Template Wizards. After deciding which output channel – print, email or web – 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 the three types of output channels, or **contexts** as they are called in the Designer; see "Creating an Email template with a Wizard" on page 425, "Creating a Print template with a Wizard" on page 423 and "Creating a Web template with a Wizard" on page 429.

After creating a template you can add the other contexts (see "Contexts" on page 218), as well as extra sections (see "Sections" on page 389), to the template.
It is, however, not possible to use a Template Wizard when adding a context or section to an existing template.

> **Tip**
>
> If an Email context is going to be part of the template, it is recommended to start with an Email Template Wizard; see "Creating an Email template with a Wizard" on page 425. After creating a template, contexts can be added to it, but that can not be done with a wizard.

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 for the first time, select **File > Save as**. After that you can save the template by selecting **File > Save** or pressing **Ctrl**+**S**.

When more than one resource (template or data mapping configuration) 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.

Sharing a template

To share a template, you can send the template file itself, or save the template to a package file, optionally together with a Data Mapping Configuration, a Job Creation Preset and an Output Creation Preset.

To create a package file, select **File > Send to Workflow** and choose **File** in the **Destination** box. For the other options, see "Sending files to Workflow" on the facing page. The package file has the extension .OL-package.

Generating output from the Designer

Output can be generated directly from the Designer; see "Generating Print output" on page 306, "Generating Email output" on page 317 and "Generating Web output" on page 320.

Sending files to Workflow

Workflow can generate output from a template as well. For this, the template has to be sent to Workflow.

The Send to Workflow dialog sends templates, Data Mapping Configurations and print presets to the Workflow server, or saves them as a package file. 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 "Job Creation Presets" on page 577 and "Output Creation Settings" on page 583 for more details.

To send one or more templates to Workflow:

1. Select **File > Send to Workflow**.
2. Select the template to send. By default the currently active template is listed. Click **Browse** to select another template. You may select more than one template in the Browse dialog, and each of them is sent to Workflow (or added to a package file). A template file has the extension .OL-template.
3. Select the Data Mapping Configuration to send. By default the current configuration is listed. Click **Browse** to select another configuration. You may select more than one configuration file in the Browse dialog, and each of them is sent to Workflow (or added to a package file). A Data Mapping Configuration file has the extension .OL-datamapper.
4. Use the drop-down to select a Job Creation Preset to send. Click **Browse** to select a preset that is not in the default location for presets. A Job Creation Preset file has the extension .OL-jobpreset.
5. Use the drop-down to select an Output Creation Preset. Click **Browse** to select a preset that is not in the default location for presets. An Output Creation Preset file has the extension .OL-outputpreset.
6. Finally, choose the **Destination**: use the drop-down to select where to send the files. The option **Workflow machines** lists all the PlanetPress Workflow installations detected on the network. Select **File** to save the files as a package that can be loaded within the Workflow tool.

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 can be added later.

To create a Print template with a Template Wizard:

1. - In the **Welcome** screen that appears after startup, choose **Browse Template Wizards**.
     Scroll down until you see the Print Template Wizards. Select one of the Wizards.
   - Alternatively, on the **File** menu, click **New**, expand the **Template** folder, and then expand the **Basic Print templates** folder. Select a template and click **Next**.

   See "Print Template Wizards" below for information about the various types of Template Wizards.
2. Make adjustments to the initial settings (the options for each type of template are listed below). Click **Next** to go to the next settings page if there is one.
3. Click **Finish** to create the template.

See "Print context" on page 349 and "Print sections" on page 352 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.

Print Template Wizards

There are two 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 351.
- 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 365.
- **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 333.
- A script called Dynamic Front Image Sample. This script shows how to toggle the image on the front page dynamically. See also "Write your own scripts" on page 373.

The Wizard opens the Print section, so that you can fill it with text and other elements; see "Content elements" on page 220. 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 349 and "Print sections" on page 352 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 360.

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 349 and "Print sections" on page 352.
- 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 365.
- 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 368.
- **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 333.

The Wizard opens the Print section. You can add text and other elements; see "Content elements" on page 220.

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 to learn how to style elements.

> **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.

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 349.

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. Building an email is not like building for the web. While web browsers use standards (to a significant extent), many email clients do not. This means that email clients can, and will, interpret the same HTML and (inline) CSS in totally different ways.

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 218 and "Email templates" on page 297).

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.
   - Alternatively, on the **File** menu, click **New**, 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 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 333.
  The Invoice email template also contains a Dynamic Table; see "Dynamic table" on page

344.

- 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 or a data mapping configuration, 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 300
- 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 220, "Email context" on page 295, and "Email templates" on page 297.

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 to make borders of elements visible on the Design tab. The borders will not be visible on the Preview tab.

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. Enter a hexadecimal color code. The color value must be a valid HTML Color Name, or a valid hexadecimal color code. To get a hexadecimal color code, you could use an online color picker tool (such as w3schools' Color Picker). The color can be changed later; see "Colors" on page 413.

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** or **Invoice** email, the Email Template Wizard lets you choose:

- The subject. You can change and personalize the subject later, see "Email header settings" on page 300.
- 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. The color value must be a valid HTML Color Name, or a valid hexadecimal color code. To get a hexadecimal color code, you could use an online color picker tool (such as W3Schools' Color Picker). The color can be changed later; see "Colors" on page 413.
- 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.

In addition, for an **Invoice** email you can change the following content settings:

- **Show Welcome Message**. Check this option to insert a salutation and one paragraph with dummy text in the email.
- **Detail Table Name**. Type the name of a detail table to fill the lines of the invoice with data. In the Designer, a detail table is a field in the Data Model that contains a variable number of items (usually transactional data).

Creating a Web template with a Wizard

With the Designer you can design Web templates and output them through Workflow or as an attachment to an email when generating Email output.

Capture On The Go templates are a special kind of Web templates; see "Capture OnTheGo (COTG) templates" on page 433.

A Web Template Wizard helps you create a Web page that looks good on virtually any browser, device and screen size.

Foundation

All Web Template Wizards make use of the Zurb **Foundation** front-end framework. A front-end framework is a collection of HTML, CSS, and JavaScript files to build upon. Foundation is a **responsive** framework: it uses CSS media queries and a mobile-first approach, so that websites built upon Foundation look good and function well on multiple devices including desktop and laptop computers, tablets, and mobile phones . Foundation is tested across many browsers and devices, and works back as far as IE9 and Android 2. See http://foundation.zurb.com/learn/about.html.

After creating a Web template, the other contexts can be added, as well as other sections (see "Adding a context" on page 219 and "Adding a Web page" on page 443).

To create a Web template with a Template Wizard:

1. - In the **Welcome** screen that appears after startup, choose **Browse Template Wizards**.
     Scroll down until you see the **Foundation Web Page Starter** Template Wizards.
   - Alternatively, on the **File** menu, click **New**, expand the **Template** folder, and then expand the **Foundation Web Page Starter** folder.
2. Select a template. There are 4 types of Web Template Wizards :
   - Blank
   - Contact Us
   - Jumbotron
   - Thank You

   If you don't know what template to choose, see "Web Template Wizards" on page 432 further down in this topic, where the characteristics of each kind of template are described.
3. Click **Next** and make adjustments to the initial settings.
   - **Section**:
     - **Name**: Enter the name of the Section in the Web context. This has no effect on output.
     - **Description**: Enter the description of the page. This is the contents of a <meta name="description"> HTML tag.
   - **Top bar** group:
     - **Set width to Grid**: Check this option to limit the width of the top bar contents to the Foundation Grid, instead of using the full width of the page.
     - **Stick to the top of the browser window**: Check to lock the top menu bar to the top of the page, even if the page has scroll bars. This means the menu bar will always be visible in the browser.
     - **Background color**: Enter a valid hexadecimal color code for the page background color (see w3school's color picker) , or click the colored circle to the right to open the Color Picker.
   - **Colors** group: Enter a valid hexadecimal color code (see w3school's color picker) or click the colored circle to open the Color Picker, and pick a color for the following elements:
     - **Primary**: links on the page.
     - **Secondary**: secondary links on the page.
     - **Text**: text on the page contained in paragraphs (<p>).

- **Headings**: all headings (<h1> through <h6>) including the heading section's subhead.

4. Click **Finish** to create the template.

The Wizard creates:

- A Web context with one web page template (also called a **section**) in it. The web page contains a Header, a Section and a Footer element with dummy text, and depending on the type of web page, a navigation bar, button and/or Form elements.
- Resources related to the Foundation framework (see "Web Template Wizards" on the facing page): style sheets and JavaScript files. The style sheets can be found in the **Stylesheets** folder on the **Resources** pane. The JavaScript files are located in the **JavaScript** folder on the **Resources** pane, in a **Foundation** folder.
- A collection of Snippets in the **Snippets** folder on the Resources pane. The Snippets contain ready-to-use parts to build the web page. Double-click to open them. See "Snippets" on page 393 for information about using Snippets.
- Images: icons, one picture and one thumbnail picture. Hover your mouse over the names of the images in the **Images** folder on the **Resources** pane to get a preview.

The Wizard opens the Web section, so that you can fill it with text and other elements; see "Content elements" on page 220, "Web Context" on page 441 and "Web pages" on page 442.

Web pages can be personalized just like any other type of template; see "Variable Data" on page 333 and "Personalizing Content" on page 322.

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.

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.

Web Template Wizards

Foundation

All Web Template Wizards make use of the Zurb **Foundation** front-end framework. A front-end framework is a collection of HTML, CSS, and JavaScript files to build upon. Foundation is a **responsive** framework: it uses CSS media queries and a mobile-first approach, so that websites built upon Foundation look good and function well on multiple devices including desktop and laptop computers, tablets, and mobile phones . Foundation is tested across many browsers and devices, and works back as far as IE9 and Android 2. See http://foundation.zurb.com/learn/about.html.

Jumbotron

The name of the Jumbotron template is derived from the large screens in sports stadiums. It is mostly useful for informative or marketing-based websites. Its large banner at the top can display important text and its "call to action" button invites a visitor to click on to more information or an order form.

Contact Us

The Contact Us template is a contact form that can be used on a website to receive user feedback or requests. It's great to use in conjunction with the Thank You template, which can recap the form information and thank the user for feedback.

Thank You

The Thank You template displays a thank you message with some text and media links.

Blank web page

The Blank Web Page template is a very simple Foundation template that contains a top bar menu and some basic contents to get you started.

Capture OnTheGo (COTG) templates

With the Designer you can create Capture OnTheGo templates. COTG templates are used to generate forms for the Capture OnTheGo mobile application. For more information about this application, see the website: Capture OnTheGo.

A Capture OnTheGo form is actually just a Web form, that you could add without a wizard, but the COTG Template Wizards include the appropriate JavaScript and styles to create user-friendly, responsive forms. They are built upon the Foundation framework.

Foundation

All Web Template Wizards make use of the Zurb **Foundation** front-end framework. A front-end framework is a collection of HTML, CSS, and JavaScript files to build upon. Foundation is a **responsive** framework: it uses CSS media queries and a mobile-first approach, so that websites built upon Foundation look good and function well on multiple devices including desktop and laptop computers, tablets, and mobile phones . Foundation is tested across many browsers and devices, and works back as far as IE9 and Android 2. See http://foundation.zurb.com/learn/about.html.

After creating a COTG template, the other contexts can be added, as well as other sections (see "Adding a context" on page 219 and "Adding a Web page" on page 443).

Creating a COTG template using a Wizard

To create a COTG template with a Template Wizard:

1. - In the **Welcome** screen that appears after startup and when you click the Home icon at the top right, choose **Browse Template Wizards**. Scroll down until you see the **Capture OnTheGo Starter** Template Wizards.
   - Alternatively, on the **File** menu, click **New**, expand the **Template** folder, and then expand the **Capture OnTheGo Starter** folder.
2. Select a template. There are 8 types of Web Template Wizards:
   - **Blank**. The Blank COTG Template has some basic design and the appropriate form, but no actual form or COTG elements.
   - **Bill of Lading**. The Bill of Lading Template is a transactional template that includes a detail table with a checkmark on each line, along with Signature and Date COTG elements. Use this wizard as a way to quickly start any new Zurb Foundation based form for Capture OnTheGo.

   - **Event Registration**. The Event Registration Template is a generic registration form asking for name, phone, email, etc.
   - **Event Feedback**. The Event Feedback Template is a questionnaire containing different questions used to rate an experience.
   - **Membership Application**. The Membership Application Template is a signed generic request form that can be used for memberships such as gyms, clubs, etc.
   - **Patient Intake**. The Patient Intake Template is a generic medical questionnaire that could potentially be used as a base for insurance or clinic form.
   - **Kitchen Sink**. The Kitchen Sink Template includes a wide range of basic form and COTG form elements demonstrating various possibilities of the software.
   - **Time Sheet**. The Time Sheet Template is a single page application used to add time entries to a list. This template demonstrates the dynamic addition of lines within a COTG template, as the Add button creates a new time entry. There is no limit to the number of entries in a single page.

3. Click **Next** and make adjustments to the initial settings.
   - **Submit URL**: enter the URL where the form data should be sent. The URL should be a server-side script that can accept COTG Form data.
   - The **Title** and the **Logo** that you choose will be displayed at the top of the Form.
   - **Background color**: Enter a valid hexadecimal color code for the page background color (see w3school's color picker).
   - Enter a valid hexadecimal color code (see w3school's color picker) for the background color of the navigation bar at the top and another for the buttons on the Form.
4. Click **Next** to go to the next settings page if there is one.
5. Click **Finish** to create the template.

The Wizard creates:

- A **Web context** with one web page template (also called a **section**) in it. The web page contains an 'off-canvas' Div element, Header, a Section and a Footer element with dummy text, and depending on the type of web page, a navigation bar, button and/or Form elements.
- Style sheets and JavaScript files related to the COTG form itself and others related to the Foundation framework (see above). The style sheets can be found in the **Stylesheets** folder on the **Resources** pane. The JavaScript files are located in the **JavaScript** folder on the **Resources** pane.

- A collection of Snippets in the **Snippets** folder on the Resources pane. The Snippets contain ready-to-use parts to build the web page. Double-click to open them. See "Snippets" on page 393 for information about using Snippets.

The Wizard opens the Web section, so that you can fill the Capture OnTheGo form.

6. Make sure to set the action and method of the form. The **action** of a Capture OnTheGo form should specify the Workflow HTTP Server Input task that receives and handles the submitted data. The action will look like this: **http://127.0.0.1:8080/action** (8080 is Workflow's default port number; 'action' should be replaced by the HTTP action of that particular HTTP Server Input task).
   The **method** of a Capture OnTheGo form should be **POST** to ensure that it doesn't hit a data limit when submitting the form. The **GET** method adds the data to the URL, and the length of a URL is limited to 2048 characters.

Adding elements to a COTG template

To fill a Capture OnTheGo form, you can use special Capture OnTheGo Form elements, such as a Signature and a Barcode Scanner element. To insert an element, select **Insert > Capture OnTheGo Form elements**and click the respective element; see "COTG Elements" on page 290. Web Form elements can also be used on COTG Forms (see "Forms" on page 266 and "Form Elements" on page 269) as well as text, images and other elements (see "Content elements" on page 220).

> **Tip**
>
> If the COTG Form replaces a paper form, it can be tempting to stick to the original layout. Although that may increase the recognizability, it is better to give priority to the user-friendliness of the form. Keep in mind that the COTG form will be used on a device and don't miss the chance to make it as user-friendly as possible.

Capture OnTheGo templates can be personalized just like any other type of template; see "Variable Data" on page 333 and "Personalizing Content" on page 322.

> **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.

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.

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 PReSDesigner

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 425. 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.

Emmet

Emmet is a plugin that enables the lightning-fast creation of HTML code though the use of a simple and effective shortcut language resembling CSS Selectors. On the Source tab in the Workspace, you could for example type **table>tr>td*2** and press the **Tab** key to quickly enter a table with one row and two cells in that row. Typing **table.green>tr*4>td*2** and pressing the **Tab** key inserts a table that has the class green, and that contains four rows with two cells each.

To learn more about Emmet, please see their website: Emmet.io and the Emmet.io documentation: http://docs.emmet.io/.

To change the way Emmet works in the Designer, select **Window > Preferences**, and in the Preferences dialog, select **Emmet**.

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, the Designer converts all CSS rules that apply to the content of the email to inline style tags, as if local formatting was applied.

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 - you could set up a process in Workflow to serve images and track open rates) 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.

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.

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 276, "Snippets" on page 393, "Styling templates with CSS files" on page 396 and "Fonts" on page 416.

Resource files can also be dragged or copy/pasted **out** of the the application to save them on a local hard drive.

Once 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 383 and "Write your own scripts" on page 373.

> **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).

Resources can also be called from a PReS Workflow instance:

- "Static Resources", as set in the preferences, are accessed using the resource path, by default something like http://servername:8080/_iRes/images/image.jpg. (For guidance on setting the preferences, search for 'HTTP Server Input 2' in the PReS Workflow help files on OL Help).
- Resources can also be served by processes: http://servername:8080/my_process?filename=image.jpg (assuming "my_process" is the action in the HTTP Server Input).

Web

With the Designer you can create one or more Web templates and merge the template with a data set to generate personal web pages.

The **Web** context is the Web output channel and the folder in the Designer that can contain one or more Web templates. CaptureOnTheGo templates are a special kind of Web templates. They are stored in the Web folder as well.

The Web context outputs an HTML web page that contains the HTML text and all the resources necessary to display it.
JavaScript files are added to the <head> in the generated HTML file. They are useful to add special features such as those offered by jQuery and its plugins, or MooTools.
Stylesheets are also added to the <head> and are used just as they would be used in a regular web page.

It is advisable to follow design guidelines for web pages, so that they are likely to look good in different browsers and on different devices and screen sizes. When you start with a Web Template Wizard, the Foundation framework is added to the template, to guarantee just that; see "Creating a Web template with a Wizard" on page 429 and "Capture OnTheGo (COTG) templates" on page 433.

When a Web template is created, either with a Wizard or by adding the Web context to an existing template (see "Adding a context" on page 219), the Web context folder is created along with other files that are specific to an Web context; see "Web Context" below.

Many of the content elements that are available for all three contexts are particularly suitable for web pages; see "Content elements" on page 220. Web templates are personalized just like any other template; see "Variable Data" on page 333.

Only one Web section is created at the start, but you can add as many Web sections as you need; see "Web pages" on the facing page. Note that when the Designer merges a data set to generate output from the Web context, it can merge only one of the templates with each record; see "Generating Web output" on page 320.

Web Context

In the Designer the Web context is the folder that contains Web page templates.

When a Web template is created (see "Creating a Web template with a Wizard" on page 429) or when a Web context is added to a template (see "Adding a context" on page 219) the following happens:

- The Web context is created and one Web page or **section** is added to it. You can see this on the **Resources** pane: expand the **Contexts** folder, and then expand the **Web** folder. See "Web pages" below to learn how to fill a web page template in the Designer. Although only one web page can be generated per record when generating Web output, the Web context can contain multiple sections. One section is created at the start, but you can add more; see "Adding a Web page" on the next page.
- A style sheet, named context_web_styles.css, is added to the template. If a Template Wizard was used to create the template, Foundation style sheets are 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 Web context; see "Styling and formatting" on page 395.

When the template is ready, you can:

- Output the web page as an as an integral HTML file attached to an Email context in the same template.
- Output the Web context in an automated Workflow using the Create Web Content task.

See "Generating Web output" on page 320

The Web context outputs an HTML web page that contains the HTML text and all the resources necessary to display it.
JavaScript files are added to the <head> in the generated HTML file. They are useful to add special features such as those offered by jQuery and its plugins, or MooTools.
Stylesheets are also added to the <head> and are used just as they would be used in a regular web page.

Web pages

Web pages (also called Web **sections**) are part of the Web context (see "Web Context" on the previous page) in a template.

The Web context outputs an HTML web page that contains the HTML text and all the resources necessary to display it.
JavaScript files are added to the <head> in the generated HTML file. They are useful to add special features such as those offered by jQuery and its plugins, or MooTools.

Stylesheets are also added to the <head> and are used just as they would be used in a regular web page.

A Web context can contain multiple templates. When generating output from the Web context, however, only one of the Web templates can be merged with each record. Set the 'default' Web section (see "Setting a default Web page for output" on page 445) before generating Web output; also see "Generating Web output" on page 320.

Creating a Web page

When creating a Web page, it is advisable to follow design guidelines for web pages, so that they are likely to look good in different browsers and on different devices and screen sizes. When you start with a Web Template Wizard, the Foundation framework is added to the template, to guarantee just that; see "Creating a Web template with a Wizard" on page 429. Other approaches are described below, in "Adding a Web page" below.

Adding a Web page

When a Web template is created (see "Creating a Web template with a Wizard" on page 429), only one Web section is added to it. A Web context may contain various templates, but per record only one of those can be used to generate output.

It is not possible to add a Web section to an existing Web context with the help of a Template Wizard.

To provide alternative content for the web page, you could use Conditional Content (see "Showing content conditionally" on page 341), or Snippets and a script (see "Snippets" on page 393 and "Loading a snippet via a script" on page 383).

> **Tip**
>
> For an example of how to serve different web pages using snippets, see the following how-to: Creating a multi-page Web template.

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 390).

To add a blank section to the Web context:

- On the **Resources** pane, expand the **Contexts** folder, right-click the **Web** folder, and then click **New Web page**.

Deleting a Web page

To delete a Web section:

- On the **Resources** pane, expand the **Contexts** folder, expand the **Web** 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. After closing and reopening the template it is no longer possible to restore the deleted context this way.

Filling a Web page

Many of the content elements that are available for all three contexts are particularly suitable for web pages; see "Content elements" on page 220. Do not use Positioned Boxes and Tables to position elements, however; use Inline Boxes instead.

Forms and Form elements are only available in a Web context; see "Forms" on page 266 and "Form Elements" on page 269.

Using variable data on a Web page

Web templates are personalized just like any other template; see "Variable Data" on page 333. There are a few extra possibilities, though: variable data can be used in Form elements and they can be passed to client-side JavaScript.

Using Variable Data in Form elements

Variable data may be used in form elements, such as a drop-down list (a Select element). How to do that, is described in this how-to: Dynamically add options to a dropdown.

Passing Variable Data to client-side JavaScript

When serving Web pages using Workflow, the HTML is first personalized and then served to the web browser by a Workflow process. At that stage custom JavaScripts do not have access to the information stored in the Data Model. To enable a client-side script to use variable data, you need to create a Text Script that produces a JSON string and stores that in the attribute of an HTML element, the `value` attribute of a hidden field for example. The custom JavaScript can than retrieve that information and use it to create dynamic page elements. Producing a JSON string and storing the results in the attribute of an HTML element are both options in the Text Script wizard; see "Using the Text Script Wizard" on page 335.

Styling and formatting a Web page

The contents of a Web section can be formatted directly, or styled with Cascading Style Sheets (CSS). See "Styling and formatting" on page 395.

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. Choose which CSS files should be applied to this section. You can also change the order in which the CSS files are read. This can have an effect on which CSS rule is applied in the end.

Setting a default Web page for output

When generating output from the Web context, only one of the Web templates can be merged with each record.

To select the Web section that will be output by default:

- On the **Resources** pane, expand the **Web** context, right-click a section and click **Set as Default**.

Tip

Use a Control Script to dynamically select a Web section for output depending on the value of a data field. See "Control Scripts" on page 385.

Including JavaScript files

Which JavaScript files are included in the a Web section, depends on a setting for that section. To change this:

1. On the **Resources** pane, right-click a section in the **Web** context and click **Includes**.
2. Choose which JavaScript files should be included in this section.

For more information about using JavaScript files, see "Using JavaScript" on the next page.

Setting the title, meta data and a shortcut icon

Each Web section has a set of properties to define the title of the web page, the shortcut icon and the meta tags appearing in the web page's head (with the HTML tag: <head>, see http://www.w3schools.com/tags/tag_head.asp).

To change these properties:

1. On the **Resources** pane, expand the **Web** context, right-click the section and click **Properties**.
2. Enter the **Page Title**. The contents of this field will go in the <title> HTML tag. (**Name** is the name of the section in the Web context; this has no effect on output.)
3. Add a **Shortcut Icon** by entering the path to the favicon.ico file, for instance images/favicon.ico.

   Tip

   If a valid favicon image is dragged to the Web section, it will automatically be set as a shortcut icon.

4. The **Meta Information Group** lists all <meta> tags that will be added to the header of the HTML file generated in the output. Click the **Add** button to add a new <meta> tag to the list. Then you can select the type of <meta> tag, which is either name or http-equiv, and

enter the value (for a name-type meta tag) or the content (for a . For more information on <meta> tags, see W3Schools - HTML meta tag.

Adding information to the <head> via script

When generating Web output, the Designer automatically adds the included resources to the <head>. To add other tags to the <head>, such as a <base> tag to set a default base URL/target for all relative URLs in a document, you need to write a script. If you are not familiar with scripts, see "Write your own scripts" on page 373 for an explanation of how scripts work.

1. Create a script: 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.
3. Choose the option **Selector** and in the **Selector** field, type `head`.
4. Write a script that appends an element to the <head> of the web page.

Example

The following script adds a <base> element to the head of a web page.

```
results.append("<base href='http://www.w3schools.com/images/'
target='_blank'>");
```

Using JavaScript

JavaScript files, libraries and frameworks can be added to a template, primarily for use in Web pages and Capture OnTheGo Forms. Before doing this, you need to choose which kind of library or framework you want to work with, depending on the type of features you really desire. For a bit of help with that and a few examples, see this how-to: Using external libraries.

Adding JavaScript files to the resources

To add a JavaScript file to the resources:

- Right-click the **Javascript** folder on the **Resources** pane, and click **New Javascript**. Double-click it to open and edit it.
- Alternatively, drag and drop the JavaScript file from the Windows Explorer to the JavaScript folder on the Resources pane.

Next, include it in a Web page; see below.

Adding a remote JavaScript file

A Remote Javascript Resource is a file that is not located within your template but is hosted on an external web server (generally called a **CDN**). When generating Web output, these files are referenced in the web page's header and are served by the remote server, not by the Connect Server module.

There are a few advantages to using remote resources:

- 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.

To add a remote javascript:

1. Right-click the **Javascript** folder on the **Resources** pane, and click **New Remote Javascript**.
2. Enter a name for the file as it appears in the Javascript resources. For better management, it's best to use the same filename as the remote resource.
3. Enter the URL for the remote resource. This must be a full URL, including the http:// or https:// prefix, domain name, path and filename.
4. Optionally, check **defer** or **async** to add the async or defer attribute to the <link> element in the <head> of the section.
   Defer postpones the execution of the script until the page has finished parsing. This attribute is required by APIs like Google Maps.
   When **async** is checked, the script executes asynchronously with the rest of the page (while the page continues the parsing).
   When neither option is checked, the script is fetched and executed immediately, while the parsing of the page is paused.
6. Optionally, for a Capture OnTheGo Form, you can check **Use cached Capture OnTheGo resource**, to prevent downloading a remote JavaScript file again if it has been downloaded before. The file should be available on a publicly accessible location, for example: a folder location on a corporate website, hosted by a CDN (Content Delivery Network) or shared via a Workflow process.

Note

In Workflow, when using the Create Web Contents task, check the **Embed All Resources** option to download and embed all remote resources.

Popular hosted frameworks on CDN networks are:

- jQuery on MaxCDN
- Zurb Foundation on CDNJS
- Bootstrap on MaxCDN
- Multiple frameworks on Google Developers

Including a JavaScript file in a Web context

To link a JavaScript file to a certain web page template:

1. On the **Resources** pane, expand the **Contexts** folder, and then expand the **Web** context.
2. Right-click a Web page and click **Includes**.
3. Check the JavaScript files that should be included with the Web page. Using the **Up** and **Down** buttons you can change the order of the files, too.
4. Click **OK**.

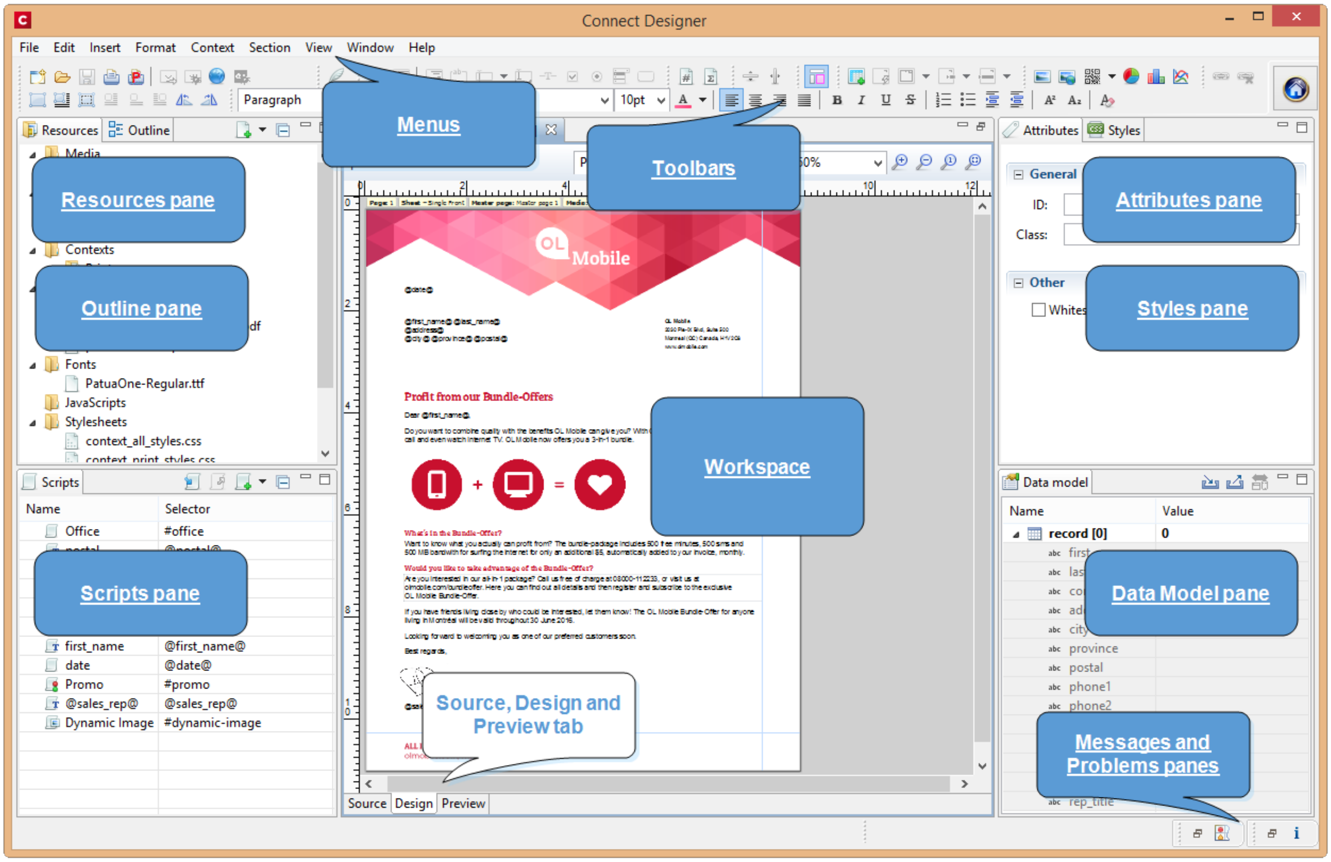
Using JavaScript in other contexts

Email clients do not support JavaScript. Therefore, Email contexts cannot include JavaScript resources.

When a JavaScript file is included in a Print section, the Designer itself acts as the browser. When Print output is generated, it runs the JavaScript after the main page flow contents and the pagination are generated. So, it is possible to change the Print output by a JavaScript; you could, for example, add a barcode that includes the pagenumber to each document. A warning is appropriate, however: changing the DOM may change the page flow and doing so at this point may result in bad output and/or serious errors or a crash of the software.

Designer User Interface

The Designer's user interface gives you several options to work with.



See:

- "Menus" on page 479
- "Toolbars" on page 519
- "Resources Pane" on page 496
- "Outline Pane" on page 495
- "Attributes Pane" on page 488
- "Styles pane" on page 517
- "Workspace" on page 517
- "Data Model Pane" on page 490
- "Scripts Pane" on page 500
- "Problems and messages" on page 493

Dialogs

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.

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** or **CMYK**. This determines how the color is saved in the formatting properties, and how they are printed or output; see "Colors" on page 413.
- **RGB group**: Enter the Red, Green and Blue color values from 0 to 255.
- **HEX**: Enter a valid hexadecimal color code; see HTML Hex Color.
- **CMYK group**: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 their equivalent.

Colors Properties

The Colors Properties defines and sets named colors used in the template; see "Colors" on page 413. Named colors can be used throughout the templates, in all contexts and are visible both in color selector dialogs and useable with their names in style sheets; see "Styling and formatting" on page 395.

- **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.
- **Edit**: Edit the currently selected color using the **Edit Color** dialog.
- **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 dialog

Email Context Properties

This dialog defines options used when generating email output.

- **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.

Print Context Finishing Options

This dialog defines finishing options for this context when it is printed. These options affect the context as a whole including all sections.

For an explanation of all Binding and Hole making options, see "Finishing Options" on page 578.

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 Formatting).
- **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.

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.

When replacing text in the Design tab, formatting in the replaced text will be removed. If formatting is necessary in the new text, it is better to use the Source tab to include the HTML yourself.
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**: [tbd]
  - **Incremental**: [tbd]
  - **Regular expressions**: [tbd]
- **Find**: Click to find the next instance of the source string.
- **Replace/Find**: [tbd]
- **Replace**: [tbd]
- **Replace All**: Click to replace all instances of the source with the target text.
- **Close**: Close the dialog.

Fonts Dialog

The Fonts dialog contains the fonts that were added to the template manually. It essentially lists the fonts located in the **Fonts** folder of the Resources Pane.

When fonts are added to the Fonts folder, the following happens:

- An entry is added to this dialog
- If more than one font has the same filename with a different extension, they are considered variations of the same fonts, shown in the Edit Font dialog.

For example, if adding 3 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 the font should be referred to as. Equivalent to the font-family property of the @font-face CSS function.
- **Font Weight**: Use the drop-down to select the default font weight ("boldness"):
  - **None**: Does not define the property.
  - **Normal**: Defines font-weight as "normal"
  - **Bold**: Defines the font-weight as "bold" (equivalent to 700).
  - **Numerical values**: Defines the line thickness. 400 is normal, 700 is bold.
- **Font Style**: Use the drop-down to select the font style:
  - **None**: Does not define the property.
  - **Normal**: Defines font-style as "normal"
  - **Italic**: Makes the font italic.
  - **Oblique**: Makes the font oblique (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.

Locale Settings

The Locale dialog box sets the locale used inside the template's record. Locale can affect time, currency output, and other formatting that depends on location and language.

This setting is specific to each template, changing it for one template will not affect the other ones.
For example. Assuming a "Date" field containing the following date: **4/11/12**, with this field being added to the template using the Text Script Generator with the "Long Date" format. If the locale is set to "**en-US**", the date appears on the page as **April 11, 2012**. Setting the locale to **fr-CA** makes this text appear as **11 avril 2012**. Setting it to zh-CN will print **2012年4月11日** .

- **Use**: Use the drop-down to select how the locale is set for the current record.
  - **System Locale**: Select to use the operating system's locale settings. This is set in the "Region" settings of the control panel.
  - **Explicit Locale**: Select to specify a static locale which will remain static for this template, whichever server the template is used on.

    - **Data Field**: Select to use a data field from the record. The locale will be record-specific in this case.
- **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. This field must be a string and contain 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 both ISO-639-1 alone ("en", "fr", etc) or ISO-639-1 followed by a 2-letter country code ("de-DE", "zh-CN", "fr-CA", "fr-FR", etc).

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 tabs controls certain aspects of the software.

General preferences

The General Preferences are as follows:

- **Always run in background**: This option correlates with the "Always run in background" option selectable in the "Document Boundaries Refresh" dialog and "Print via Print Server" dialog. When either of these dialogs is used and the option is checked, it will also be checked here. To prevent the refresh boundaries and print via print server dialogs to automatically run as background, uncheck this option.

Cleanup Service preferences

The Cleanup Service defines how entities in the Connect Server Module are cleaned up when they have been left unused. The values below only define when the specified objects are *set* for deletion, not when they are actually deleted. Items are only deleted when the cron job runs on its schedule, or when the Run Now button is pressed in this dialog.

The more items are present in your database, and the larger they are, the more time and processing power (CPU) will be used to clean them up. It is recommended to clean up the database as often as possible, especially if you are not retrieving items from the database at a later date. You may also set the database cleanup outside of business hours (see the cron schedule option below).

- **Enable cleanup service**: Check to enable the cleanup service Uncheck to disable it.
- **Run at application start up**: Click to start the service when the Designer or DataMapper module is opened.
- **Run according to the cron schedule**: Enter the interval at which the cleanup service runs. To know how to write the schedule, please refer to the Quartz Scheduler reference page.
- **Object Retention group**:
  - **Minimum time to retain Data Sets**: The minimum time during which a Data Set (and all its contained records) is kept in the database before being set for deletion.
  - **Minimum time to retain Content Sets**: The minimum time during which a Content Set (and all its contained content items) is kept in the database before being set for deletion.
  - **Minimum time to retain Job Sets**: The minimum time during which a Job Set (and all its contained jobs) is kept in the database before being set for deletion.
  - **Minimum time to retain Managed Files**: The minimum time during which files such as data mapping configurations and templates are kept in the database before being set for deletion.
  - **Allow orphan file cleanup**: Check to automatically detect orphan files and set them for deletion. Orphan files can be resources, and internal files used by Connect.
  - **Minimum time to retain orphaned temp files**: The minimum time during which orphaned files are kept in the database before being set for deletion.
  - **Allow orphan database entity cleanup**: Check to automatically detect orphan database entities and set them for deletion. Orphaned database entities are entries in the database that are no longer being referred to by any other entity (for example a record that is not part of a record set).
  - **Minimum time to retain orphaned database entries**: The minimum time during which orphaned database entities are kept in the database before being set for deletion.

Note

Only a server that has the MySQL module installed should run any of these options, with the exception of Allow orphan file cleanup. This option refers to local files and should be run on each machine in the cluster (however, not at the same time). See "Server Clustering" on page 34.

Datamapper preferences

Datamapper XML Preferences

- **Display New Line Character as ¶** : Check to show line returns as ¶ in the Data Viewer, when XML files are shown. If the option is unchecked, you will not see spaces and line returns after element names in the Data Viewer.

Datamapper Default Format Settings

Users do not always process data files that were generated on their own system, therefore those settings rarely match. So instead of specifying these settings for every single field of that type, which can become cumbersome, Default Format Settings can be defined at the user level, at the Datamapper configuration level and/or at the field level. So, remember that:

- Datamapper stores user preferences for the Date, Number and Currency formats.
- By default, the user preferences are set to the system preferences
- These user preferences become the default format values for any newly created Datamapper Configuration.
- The Datamapper can store different preferences for any specific Datamapper Configuration. These Configuration based values are used for any newly created field extraction.
- Any current format already defined for an existing field remains untouched.
- The existing preference specified in an existing field is always used, regardless of the User or Configuration preferences.
- The Configuration-wide preference is always assigned by default to any newly created field.
- The User-wide preference is always assigned by default to any new Datamapper Configuration
  - **Negative Sign Before** : A negative sign will be displayed before any negative value.
  - **Decimal Separator** : Set the decimal separator for a numerical value.
  - **Thousand Separator** : Set the thousand separator for a numerical value.
  - **Currency Sign** : Set the currency sign for a currency value.
  - **Date Format** : Set the date format for a date value.
  - **Date Language** : Set the date language for a date value (ex: If English is selected, the term May will be identified as the month of May).
  - **Treat empty as 0** : A numerical empty value is treated as a 0 value.

Editing preferences

These preferences define different editing options in the Designer module.

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

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.

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.
    - **Email Address**: Enter the email that is set by default in the "From Email" field in the Send Email and Send Test Email dialogs.
- **Litmus account Group**:
    - **Email Test address**: If you have a Litmus account, enter the test address to use when using the Send Test Email dialog. For more information on Litmus, please see http://litmus.com/

Email (SMTP) Preferences

This page defines SMTP server presets that can be selected when sending emails using either the Send Email or Send Test Email dialogs. For all three presets, the password is not saved and must be re-entered when sending emails.

- **Production Group**: Defines the default SMTP settings when using the Send Email dialog, which is considered "Production".
    - **Host**: The SMTP server through which the emails are to be sent. Can be a host (mail.domain.com) or an IP address.

  - **Use authentication**: Check if a username and password are needed to send emails through the host.
  - **Start TLS**: Enabled if authentication is checked. Sends emails through Transport Layer Security (TLS), which is sometimes referred to as SSL.
  - **User**: Enter the username used to connect to the SMTP server.
- **Test Group**: Defines the default SMTP settings when using the Send Test Email dialog, which is considered "Test".
  - **Host**: The SMTP server through which the emails are to be sent. Can be a host (mail.domain.com) or an IP address.
  - **Use authentication**: Check if a username and password are needed to send emails through the host.
  - **Start TLS**: Enabled if authentication is checked. Sends emails through Transport Layer Security (TLS), which is sometimes referred to as SSL.
  - **User**: Enter the username used to connect to the SMTP server.
- **Mandrillapp.com Group**: Defines the SMTP settings if using the service of Mandrill. For more information please see http://www.mandrill.com/ .
  - **Host**: The SMTP server through which the emails are to be sent. Can be a host (mail.domain.com) or an IP address.
  - **Use authentication**: Check if a username and password are needed to send emails through the host.
  - **Start TLS**: Enabled if authentication is checked. Sends emails through Transport Layer Security (TLS), which is sometimes referred to as SSL.
  - **Username**: Enter the username used to connect to the SMTP server.

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. To learn more about Emmet itself, please see their website Emmet.io and the Emmet.io documentation.

- 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**: [TBD]

  - **Extensions Path**: Choose a folder where to put extensions to Emmet, which includes custom snippets, preferences and syntax profiles. For more information see Customization.

Emmet Abbreviation Preferences

Abbreviations are the heart of the Emmet toolkit: these special expressions are expandable and transformed into structured code block, HTML for example. The abbreviation’s syntax looks like CSS selectors with a few extensions specific to code generation.

- Abbreviation List:
  - **Name**: The name of the item, aka its trigger. Disabled the checkbox next to the name to disable the item (it will not trigger).
  - **Context**: The context in which the item is enabled (HTML, CSS, etc)
  - **Description**: A short description of the item.
  - **Auto Insert**: [TBD]
  - **New**: Click to create a new item.
  - **Edit**: Click to modify the currently selected item.
  - **Remove**: Click to remove the currently selected item from the list.
  - **Restore Removed**: [Currently not functional]
  - **Revert to default**: [Currently not functional]
  - **Import**: Click to open a browse dialog to import an xml file containing exported items. The imported items are added to the list, and any
  - **Export**: Click to open a save as dialog to export all the items in an xml file that can be shared and re-imported.
- **Preview box**: Displays a preview of the expanded item when one is selected.

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.

- **Tag Case**: Defines whether tags in the expanded code will be in lowercase, uppercase, or keep whichever case was used when typing in the abbreviation. For example, the keep option means typing P>a expands to <P><a href=""></a></P>.

- **Attribute Case**: Defines whether attributes in the code will be in lowercase, uppercase, or keep whichever case was used when typing in the abbreviation. For example, the keep option means typing p[TITLE="Hello world"] expands to <p TITLE="hello world"></p>
- **Attribute quotes**: Defines whether the quotes in the expanded code are single (') or double ("). This has no effect on functionality, it's a visual preference for the code.
- **Each tag on new line**: Output each tag on new line with indentation, boolean. Values are `true` (each tag on new line), `false` (no formatting) and `'decide'` (string; only block-level elements on new lines).
- **Place caret placeholders in expanded abbreviations**: Defines whether leaf block-level node (e.g. node with no children) should have formatted line breaks inside. Only
- **Indent Tags**: [TBD]
- **How many inline elements should be to force line break**: [TBD]
- **Self-closing style for writing empty elements**:
- **Applied filters**: [TBD]

Emmet Snippets Preferences

Snippets are just blocks of plain code, just like in all programmers' editors. You can type anything there and it will be outputted "as-is", without any transformation. Snippets are very similar to Abbreviations except for their more static nature.

- Snippet List:
  - **Name**: The name of the item, aka its trigger. Disabled the checkbox next to the name to disable the item (it will not trigger).
  - **Context**: The context in which the item is enabled (HTML, CSS, etc)
  - **Description**: A short description of the item.
  - **Auto Insert**: [TBD]
  - **New**: Click to create a new item.
  - **Edit**: Click to modify the currently selected item.
  - **Remove**: Click to remove the currently selected item from the list.
  - **Restore Removed**: [Currently not functional]
  - **Revert to default**: [Currently not functional]
  - **Import**: Click to open a browse dialog to import an xml file containing exported items. The imported items are added to the list, and any
  - **Export**: Click to open a save as dialog to export all the items in an xml file that can be shared and re-imported.
- **Preview box**: Displays a preview of the expanded item when one is selected.

Emmet Variables Preferences

Variables are placeholders used in 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]`.

- Variable List:
  - **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.

Language Setting Preferences

The Language Setting Preferences are as follows:

- **Language**: Use the drop-down to select the User Interface Language to use.

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 583.

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 PReS 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

The General Print Preferences are used to set communication settings with the PReS Connect Server module that does the actual generation of print output. The Server module can be located on the same computer (hostname: localhost) or on a different machine. Multiple Designer modules can use a single Server module to generate Print output, as long as the appropriate hostname, username and password are provided. In essence, this can be used to create a single Print Server.

- **Print Server Settings group:**
  - **Protocol**: Use the drop-down to select whether to use the HTTP or the secure HTTPS protocol to connect to the Print Server.
  - **Hostname**: Enter the IP, machine name or URL of the Print Server. Default is `localhost`.
  - **Port**: Enter the port through which to communicate with the Print Server. Default is `9340`.
  - **Username**: Enter the username to authenticate to the Print Server. Default is `ol-admin`. This is set on the server's"Server Security Settings" on page 33.
  - **Password**: Enter the password to authenticate to the Print Server. Default is `secret`.
  - **Confirm Password**: Re-enter the password above.
- **External sort command timeout (seconds)**: Enter the number of seconds to wait for an external sort command before giving up. External sort commands are set up in the Sorting Options page of the Output module.

Print Measurements Preferences

- **Units**: Use the dropdown to specify the default measurements system used for dimensions of the template and boxes. In addition it defines the coordinates/position of box elements.

Scheduling Preferences

The scheduling preferences are a way to control precisely how the PReS Connect services work in the background when using server clustering; see "Server Clustering" on page 34.

Scheduling options

This preference page defines what is considered a small or large job (anything in between is considered "medium" jobs). For a detailed description of all options, see Scheduling Preferences.

Scheduling - Merge engine

This preferences page defines how different instances and speed units are attributed to different jobs when creating output documents. For a detailed description of all options, see Merge Engine Scheduling.

Scheduling - Weaver engine

This preference page determines the number of engines launched, as well as their speed, when generating Print Output of any type. For a detailed description of all options, see Weaver Engine Scheduling.

Scripting Preferences

The Scripting preferences define different options related to scripting with the PReS Connect interface.

- **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!

Web Preferences

Web Form Preferences

These preferences define the default behavior of some form elements.

The preferences are as follows:

- **Insert Form Field Defaults**:
  - **Style**: Defines how labels are added to input form elements:
    - **Wrap input with label**: The label is wrapped around the element, such as <label>First Name <input type="text" name="first_name"></label>
    - **Attach label to input**: The label is placed before the input, and refers to it: <label for="first_name">First Name</label> <input type="text" name="first_name">
    - **Use label as placeholder**: The label is removed and the text is put as a placeholder, such as: <input type="text" name="first_name" placeholder="First Name">
    - **No label**: The label value is ignored.
  - **Insertion Point**: Defines how new elements are inserted, by default:

- **At cursor position**: The element is inserted where the cursor is located in the template.
- **Before element**: The element is inserted before the current element where the cursor is located. For example if the cursor is within a paragraph, insertion occurs before the <p> tag.
- **After start tag**: The element is inserted within the current element, at the beginning, just after the start tag.
- **Before end tag**: The element is inserted within the current element, at the end, just before the end tag.
- **After element**: The element is inserted after the current element where the cursor is located. For example if the cursor is within a paragraph, insertion occurs after the <p> tag.

PDF Page selector

This dialog is accessed by clicking the Page button next to the Source box in the Attributes pane. It is used to select the page which is shown in the PDF source.

- **Source**: Read-only box displaying the PDF source file. To change the source file you must do it from the Attributes pane.
- **Page**: Enter the page number to show or use the up and down arrow to select the page.
- **Number of Pages**: Next to the page selector, a label indicates the number of pages in the PDF.
- **Page Preview**: Displays a preview of the specified page of the source.

Preflight dialog

Preflight is the act of executing the template and verifying that there will be no issues when actually generating output. Preflight, however, does not actually produce the output.

To start preflight, simply go in the Menus under Context, then Preflight. The preflight runs automatically and displays the results once it's done.

Profile Scripts

The Script Profiler is accessible through the **Context** -> **Profile Scripts**menu option. It runs the scripts in the template in order to verify the speed at which scripts in the Scripts Pane execute. It helps greatly in troubleshooting performance issues caused by scripting.

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 466).

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, 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 even 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.

Section properties dialogs

Email Section Properties

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.
- **Subject**: Enter a name for the default Subject of any email sent out. This is superseded by the Email Subject Script Generator when it is present.

Includes Tab

This tab defines what other resources are included in the output

- **Stylesheets**: What CSS stylesheets to use in producing the output. Stylesheets are loaded in the order shown, and styles in later Stylesheets overwrites earlier ones when the same selector is used.
- **JavaScript** (Web and Print only): Which JavaScript resources are included in the HTML header of the web output.

- **Up**: Move the selected StyleSheet or JavaScript up in priority.
- **Down**: Move the selected StyleSheet or JavaScript down in priority.

Print Section Properties

The Section Properties dialog is separated in a few separate tabs depending on the Context in which it resides:

General tab (all Contexts)

- **Section** group:
  - **Name**: The name of the Section.
  - **Show PDF data mapping input as background image**: Check this option to display each page of a PDF data source when using a PDF data mapping configuration. Each page of the PDF is shown, separated by the appropriate records. Note that as many pages as there are in the PDF will be created in the section.
- **Page**group:
  - **Size**: The named page size.
  - **Width**: The width of the page.
  - **Height**: The height of the page.
  - **Portrait**: Whether the page is portrait (otherwise, it's landscape).
- **Margins** group: Defines the margins where contents will not appear on the page.
  - **All sides**: Check to use the same margin setting for all sides. Note that this will copy the *Top*value to all margins, overwriting existing values.
  - **Top**: The top margin.
  - **Left**: The left margin.
  - **Bottom**: The bottom margin.
  - **Right**: The right margin.
- **Bleed group**: This group defines the bleeds for the sections added within the print context.
  - **All sides**: Check to define all bleeds identically using the **Top** value.
  - **Top**: The bleed at the top of the page.
  - **Left**: The bleed at the left of the page.
  - **Bottom**: The bleed at the bottom of the page.
  - **Right**: The bleed at the right of the page.

Includes Tab

This tab defines what other resources are included in the output

- **Stylesheets**: What CSS stylesheets to use in producing the output. Stylesheets are loaded in the order shown, and styles in later Stylesheets overwrites earlier ones when the same selector is used.
- **JavaScript** (Web and Print only): Which JavaScript resources are included in the HTML header of the web output.
- **Up**: Move the selected StyleSheet or JavaScript up in priority.
- **Down**: Move the selected StyleSheet or JavaScript down in priority.

Finishing tab

This tab defines finishing options for this section when it is printed.

- **Binding** group:
  - **Style**: What type of binding to request on the printer. This includes Stapled, Glued, Stitched, etc.
  - **Side**: On which side the binding occurs, such as Bottom, Left, top, etc.
  - **Location**: The location of the binding, such as Saddle, Side, Corner, etc.
  - **Angle**: If the binding should be done horizontally, vertically, or at an angle.
  - **Item count**: When certain binding styles are selected which require multiple items (such as Stapled or Stiched), use the Count option to specify the number of items the printer should use, or select Default to let the printer decide how many items to use.
  - **Area**: [TBD]
- **Hole making** group:
  - **Number of holes**: When certain binding styles are selected which require the printer to create holes in the paper, use this drop-down to select the number of holes that should be created, or use Default to let the printer decide.
  - **Pattern Catalog ID**: When the Number of Holes is not the default in the option above, use the drop-down to select the Pattern Catalog ID for the hole making.

Sheet Configuration Tab

This tab defines how different Print Context Sections output on different Media and using different Master Pages.

There are multiple groups, each defining the settings for individual *position* within the section as it outputs: **First**, **Middle** and **Last** sheet, as well as a group for **Single** sheets.

This tab contains the following options:

- **Duplex**: Check to enable content to be printed on the back of each sheet. Your printer must support duplex for this option to work.
- **Tumble**: Check to enable tumble mode so pages are duplexed as in a notebook (on Portrait output, this would be equivalent to short-edge duplex).
- **Same for all positions**: Check to enable a single group below, which defines the same options for all positions in the document. If unchecked, individual position options are available.

The next options are identical for all positions, but of course only affects the position where you change the options:

- **Allow content on**: Selects on which face of the sheet content is allowed. If "Front only" or "Back Only" is selected, 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.
- **Media**: Defines the media that is used. If the Media has *Preprinted Media*defined, the selected preprinted media image is shown as a background to each page that correspond to the media's sheet position.
  - **Edit Script**: Click to open a Script Editor dialog. The script defines what Media is used, so it can be dynamically defined using data from the source record or the extracted record.
- **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 front of the selected sheet's position. Disabled if "Back Only" is selected under Allow content on, or if Duplex is unchecked.

Background Tab

This tab defines the background for the current Print Context Section. It contains the following options:

- **General** group:
  - **PDF**: Select the PDF to use as the section's background: a PDF Datamapper Input or a PDF Resource.
  - **Path**: The path to a PDF Resource. Enter a path and file name or click the Browse button to open the Select Image dialog; see "Select Image dialog" on page 474.

- **Position**:
  - **Centered**: The PDF will be centered on the page and will not be resized.
  - **Fit to Media**: The PDF will be resized to fit the Media.
  - **Absolute**. Set a position for the top left corner of the PDF. The PDF will not be resized.
  - **Top**: the distance between the top of the page and the top of the PDF
  - **Left**: the distance between the left side of the page and the left side of the PDF.
- **Pages** group:
  - **All**: Use all pages in the PDF file as the section's background. For each page in the PDF file one page will be added to the Print section.
  - **Pages**: Select a number of pages from the PDF file. For each page in the range, one page will be added to the Print section.

Web Section Properties

The Web Section Properties defines some of the web page properties, especially details appearing in the header.

Properties Tab

- **Section Group**:
  - **Name**: Enter the name of the Section in the Web Context. This has no effect on output.
  - **Page Title**: Enter the title for the page. This is the contents of the <title> HTML tag.
  - **Shortcut Icon**: Enter the path to the favicon.ico file, for instance *images/favicon.ico*. If a valid favicon image is dragged to the Web Section, it will automatically be set as a shortcut icon.
- **Meta Information Group**: This lists all <meta> tags that are added to the header of the HTML file generated in the output. For more information on <meta> tags, see W3Schools - HTML meta tag.
  - **Type**: Select the type of <meta> tag. This is either *name*or*http-equiv*.
  - **Value**: Enter the value of the <meta> tag, for instance when *name*is selected, this could be*keywords* or*description*.
  - **Content**: Enter the desired contents of the <meta> tag.
  - **Add**: Click to add a new <meta> tag. to the list.
  - **Delete**: Click to delete the currently selected <meta> tag.

- **Move up**: Click to move the currently selected <meta> tag up one position.
- **Move down**: Click to move the currently selected <meta> tag down one position.

Includes Tab

This tab defines what other resources are included in the output

- **Stylesheets**: What CSS stylesheets to use in producing the output. Stylesheets are loaded in the order shown, and styles in later Stylesheets overwrites earlier ones when the same selector is used.
- **JavaScript** (Web and Print only): Which JavaScript resources are included in the HTML header of the web output.
- **Up**: Move the selected StyleSheet or JavaScript up in priority.
- **Down**: Move the selected StyleSheet or JavaScript down in priority.

Arrange Sections

The Arrange dialog is used to change the order of sections within a context. To access the Arrange dialog, 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.

Section Includes

This dialog defines what other resources are included in the output

- **Stylesheets**: What CSS stylesheets to use in producing the output. Stylesheets are loaded in the order shown, and styles in later Stylesheets overwrites earlier ones when the same selector is used.
- **JavaScript** (Web and Print only): Which JavaScript resources are included in the HTML header of the web output.
- **Up**: Move the selected StyleSheet or JavaScript up in priority.
- **Down**: Move the selected StyleSheet or JavaScript down in priority.

Send to Workflow/Files dialog

The Send to Workflow dialog sends templates, data mapping configurations and print presets to the PReS Workflow server, or saves it as a package file. Note that package files cannot be loaded from PReS Workflow at the moment.

- **Files to Package** group:
    - **Template**: Select the template to send. By default the currently active template is listed. Click Browse to select another template. In version 1.3 you may select more than one template in the Browse dialog, and each of them is sent to Workflow or added to a package file.
    - **Data mapping configuration**: Select the data mapping configuration to send. By default the current configuration is listed. Click Browse to select another configuration. In version 1.3 you may select more than one configuration file in the Browse dialog, and each of them is sent to Workflow or added to a package file.
    - **Job Creation Preset**: Use the drop-down to select a Job Creation Preset to send. Click Browse to select a preset that is not in the default save location.
    - **Output Creation Preset**: Use the drop-down to select an Output Creation Preset. Click Browse to select a preset that is not in the default save location.
- **Destination** group:
    - **Send files to**: Use the drop-down to select where to send the files.
        - **Workflow machines**: Send the files to a PReS Workflow installation. This lists all the detected PReS Workflow installations detected on the network.
        - **File...**: Click to save the files as a package. This package can be loaded within the Workflow tool.

Select Image dialog

The Select Image dialog lets you select an image, depending on where the image is located.

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 an image file that resides in a folder on a hard drive that is accessible from your computer. A preview of the selected image will be shown below.
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:/resources/images/image.jpg.
Browse: opens an explorer window to browse folders and select an image.

Url lets you select an image file from a specific web address. Select the protocol and then enter the URL (for example, http://www.mysite.com/images/image.jpg). A preview of the selected image will be shown below.
Protocol: **http** or **https**.

Save with template: inserts the file 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.

Stylesheets dialog

The Stylesheet editor dialog is used to edit CSS Stylesheet resources. For the time being, only two contexts are editable: Global (context_all_styles.css , which by default applies to all contexts) and Print (context_print_style.css , which by default applies to the print context only).

- **Context**: Use the drop-down to select a specific context such as *Print* or *Global* (all contexts). 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 stylesheet.
- **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 facing 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 facing 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" on the facing page.
- **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 stylesheet 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 and how it affects all elements subject to that selector.

At any point you can click on the **Advanced** button to see the Advanced Stylesheet Rule. See

- **Name**: The CSS Selector to which this rule applies. See CSS Selectors.

Type Tab

- **General group**:
  - **Font**: Select the font used to display text, equivalent to the `font-family` property.
  - **Size**: Enter the size in measure, named size or percentage, equivalent to the `font-size` property.
  - **Color**: Select a named font color as defined in the Colors Editor, create a new color or enter a color manually for text to be displayed. The color value must be a valid HTML Color Name, or a valid HTML Hex Color. Equivalent to the `color` property.
  - **Background Color**: Select a named font color as defined in the Colors Editor, create a new color or enter a color manually for the background color of the element. The color value must be a valid HTML Color Name, or a valid HTML Hex Color. Equivalent to the `backround-color` property.
- **Spacing group**:
  - **Letter Spacing**: Set the space between characters in a text in measure or percentage. Equivalent to the `letter-spacing` property.
  - **Word Spacing**: Set the space between each word in a text in measure or percentage. Equivalent to the `word-spacing` property.
  - **Whitespace**: Specify how to handle white spaces inside of an element. See CSS White-Space for details. Equivalent to the `white-space` property.
- **Style group**: Check any option to apply the selected style to text within the element:
  - **Bold**: Sets `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-transformtolowercase.
    - **Small-caps**: Sets font-varianttosmall-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-heightproperty.
    - **Align**: Select how text should be aligned, such asleft,center,rightorjustify. Equivalent to thealignproperty.
    - **First Indent**: Specify the indentation of the first line of each paragraph in the element. Equivalent to thetext-indentproperty.
    - **Display**: Select how to display the element. This can also be used to hide an element completely using thenoneoption. See CSS Display. Equivalent to thedisplayproperty.
- **Breaks group**:
    - **Before**: Specifies how to handle page breaks before the element. Equivalent to the page-break-before property.
    - **Inside**: Specifies whether to accept page breaks within the paragraph. Equivalent to the page-break-inside property.
    - **After**: Specifies how to handle page breaks after the element. Equivalent to the page-break-after property.
    - **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.

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 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

- **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.

Menus

The following menu items are shown in the Designer menu:

File Menu

- **New...**: Opens the **New (Select a Wizard)** dialog. You can choose from the Email, Print or Web Template Wizards. See "Templates" on page 420.
- **Open**: Opens a standard File Open dialog. This dialog can be used to open Templates and Data Mapping Configurations.
- **Open Recent**: List the most recently opened Templates and configurations. Clicking on a template will open it in the Designer module, clicking on a Data Mapping Configuration will open it in the DataMapper module.
- **Close**: Close the currently open Data mapping configuration or Template. If the file needs to be saved, the appropriate Save dialog will open.
- **Close All**: Close any open Data Mapping Configuration or template. If any of the files needs to be saved, the Save Resources dialog opens.
- **Close Others**: Close all Data mapping configuration and templates except the one that is currently active in the workspace.
- **Save**: Saves the current Data mapping configuration or Template to its current location on disk. If the file has never been saved, the Save As dialog appears instead.
- **Save All**: Saves all open files. If any of the open files have never been saved, the Save As dialog opens for each new unsaved file.
- **Save As...**: Saves the current file to a new location on disk.
- **Revert**: Appears only in the Designer module. Reverts all changes to the state in which the file was opened or created.
- **Add Data**: Adds data either to the current data mapping configuration or to the open template. See "Loading data" on page 324 .
    - **From File Data Source...**: Opens the dialog to add a new data file to the currently loaded data mapping configuration. Not available if the currently loaded data mapping configuration connects to a database source.
    - **From Database Data Source...**: Opens the Edit Database Configuration dialog. Not available if the currently loaded data mapping configuration is file-based.
    - **Generate Counters**: Opens the Generate Counter Wizard to create a custom counter as a data source.
- **Send to Workflow...**: Opens the "Send to Workflow/Files dialog" on page 474 to send files to a local Workflow software installation.
- **Print**: Opens the "Print Options" on page 583 dialog.

- **Print Presets**: Selecting this option allows you to create or modify Printing Presets, which can be saved and used in print runs thereafter.
  - **Job Creation Presets**: Click to open the"Job Creation Presets" on page 577 dialog.
  - **Output Creation Presets**: Click to open the "Output Creation Settings" on page 583 dialog.
- **Proof Print**: Opens the "Print Options" on page 583 dialog as a "Proof Print" 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 also "Generating Email output" on page 317.
- **Send Test Email**: Click to open the Send Test Email dialog.
- **Send COTG Test**: Click to open the Send COTG Test dialog, to send the current Web Context to the Capture OnTheGo Application.
- **Exit**: Closes the software. If any of the files need 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 places it in the clipboard.
- **Copy**: Places a copy of the currently selected text, object or element in 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.
- **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 Stylesheet Editor dialog. See "Stylesheets dialog" on page 475.
- **Colors...:** Opens the Colors Editor dialog.
- **Locales...**: Opens the Locale Settings dialog.
- **Color Settings...**: Opens the Color Settings dialog.

Insert Menu

- **Image**
  - **From file...**: 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 276.

- **From address...**: Inserts an image using a URL instead of a resource, at the current location of the pointer and opens its properties. See "Images" on page 276.

- **Text**:
  - **Wrap in span**: When text is selected, wraps that text in <span> element. This span can be used for selections, conditions, styling, etc.
- **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 469) 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 en-space).
  - **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**: Click to open the "Date" on page 264 dialog to add a date to the template based on the current system's date and time.
- **Table**
    - **Standard**: Inserts a table with a specific number of columns and rows through the Standard Table Wizard; see "Table" on page 280.
    - **Dynamic**: Inserts a dynamic table where the number of rows is determined by a Details table, through the Dynamic Table Wizard; see "Dynamic table" on page 344.
- **Table Elements**:
    - **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.
- **Common Elements**:
    - **Paragraph...**: Click to open a dialog to add a <p> element; see "Text and special characters" on page 282
    - **H1 through H6...**: Click to open a dialog to add a <h1> to <h6> element; see "Text and special characters" on page 282
    - **Address...**: Click to open a dialog to add an <address> element.
    - **Preformatted...**: Click to open a dialog to add a <pre> element.
- **Structural Elements**:
    - **Div**...: Click to open a dialog to add a <div> element; see "Boxes" on page 262
    - **Span**...: Click to open a dialog to add a <span> element; see "Boxes" on page 262
    - **Article**...: Click to open a dialog to add an <article> element
    - **Section**...: Click to open a dialog to add a <section> element.
    - **Header**...: Click to open a dialog to add a <header> element.
    - **Footer**...: Click to open a dialog to add a <footer> element.
    - **Nav**...: Click to open a dialog to add a <nav> element.
    - **Aside**...: Click to open a dialog to add an <aside> element.
- **Form Elements**:

  - **Form...**: Click to open a dialog to add a Form Element; see "Forms" on page 266
  - **Fieldset...**: Click to open a dialog to add a Fieldset Element; see "Form Elements" on page 269
  - **Text Field...**: Click to open a dialog to add a Text Field; see "Form Elements" on page 269
  - **Email Field...**: Click to open a dialog to add an Email Field; see "Form Elements" on page 269
  - **URL Field...**: Click to open a dialog to add a URL Field; see "Form Elements" on page 269
  - **Password Field...**: Click to open a dialog to add a Password Field; see "Form Elements" on page 269
  - **Number Field...**: Click to open a dialog to add a Number Field; see "Form Elements" on page 269
  - **Date Field...**: Click to open a dialog to add a Date Field; see "Form Elements" on page 269
  - **Text Area...**: Click to open a dialog to add a Text Area; see "Form Elements" on page 269
  - **Hidden Field...**: Click to open a dialog to add a Hidden Field; see "Form Elements" on page 269
  - **Label...**: Click to open a dialog to add a Label; see "Form Elements" on page 269
  - **Checkbox Field...**: Click to open a dialog to add a Checkbox; see "Form Elements" on page 269
  - **Radio Button...**: Click to open a dialog to add a Radio Button; see "Form Elements" on page 269
  - **Select Field...**: Click to open a dialog to add a Select (drop-down); see "Form Elements" on page 269
  - **Button...**: Click to open a dialog to add a Button; see "Form Elements" on page 269

- **COTG Form Elements**:

  - **Signature...**: Click to open a dialog to add a Signature Element, see "COTG Elements" on page 290.
  - **Date...**: Click to open a dialog to add a Date Element, see "COTG Elements" on page 290.
  - **Formatted Date...**: Click to open a dialog to add a Formatted Date Element, see "COTG Elements" on page 290
  - **Time...**: Click to open a dialog to add a Time Element, see "COTG Elements" on page 290.

    - **Formatted Time...**: Click to open a dialog to add a Formatted Time Element, see "COTG Elements" on page 290.
    - **Geolocation...**: Click to open a dialog to add a Geolocation Element, see "COTG Elements" on page 290.
    - **Locale...**: Click to open a dialog to add a Locale Element, see "COTG Elements" on page 290.
    - **Camera...**: Click to open a dialog to add a Camera Element, see "COTG Elements" on page 290.
    - **Barcode Scanner...**: Click to open a dialog to add a Barcode Scanner Element, see "COTG Elements" on page 290.
    - **User Account...**: Click to open a dialog to add a User Account Element, see "COTG Elements" on page 290.
    - **Device Info...**: Click to open a dialog to add a Device Info Element, see "COTG Elements" on page 290.

- **Form Wizard**: Click to open the Form Wizard to add a form to a Web Context; see "Forms" on page 266
- **Validation Wizard**: Click to open the Validation Settings dialog to change the validation settings on the currently selecting tools; see "Changing a Form's validation method" on page 269
- **Business Graphics**: Displays a list of available business graphic object to be inserted:
    - **Pie Chart**: Click to insert a new Pie Chart object and open the Chart Script Wizard.
    - **Bar Chart**: Click to insert a new Bar Chart object and open the Chart Script Wizard.
    - **Line Chart**: Click to insert a new Line Chart object and open the Chart Script Wizard.
- **Barcode**: Displays a list of available barcodes. Click on one to insert it on the page. The Text Wizard is opened for the barcode data.

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 404.
    - **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 Formating dialog and checking the appropriate style.

- **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 404.
- **Align**: When an element is selected, determine how its contents is aligned in the element: Align Left, Align Right, Align Center and Justify.
- **Paragraph...**: Opens the Paragraph Formatting to modify the paragraph where the cursor is located.
- **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 using a *float:left* style.
  - **Right**: Floats the current element to the right using a *float:right* style.
  - **None**: Removes any *float* style applied to the currently selected element.
- **Box...**: Opens the Box Formatting dialog to modify the box where the cursor is located.
- **Image...**: Opens the Image Formatting dialog to modify the image that is currently selected.
- **Table...**: Opens the Table Formatting dialog to modify the table where 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 the Table Cell Formatting dialog 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 274.
  - **Edit...**: Opens the properties for the currently selected hyperlink; see "Hyperlink and mailto link" on page 274.
  - **Remove**: Removes the currently selected hyperlink. The text or element that was the hyperlink is not removed.

Context Menu

- **Add**:
  - **Print Context**: Click to add a new Print Context to the template if one does not exist.
  - **HTML Email Context**: Click to add a new Email Context to the template if one does not exist.

  - **Web Page Context**: Click to add a new Web Context to the template if one does not exist.
- **Delete**: Click to delete the currently selected context. If only one context exists, it cannot be deleted.
- **Go to**: Click to go to the selected context. This is the same as double-clicking on the first section of any context in the Resource Pane.
- **Properties**: Click to open the currently selected context's properties.
- **Preview HTML**: Click to preview the currently open Section in the default system browser to preview it. This feature works in all contexts.
- **Profile Scripts**: Click to open the Profile Scripts dialog to test script performance. The profiler runs automatically when it is open.
- **Preflight**: Click to preflight the document and open the Preflight Dialog. Preflight verifies the template for common errors.

Section Menu

- **Add**: Click to create a new section in the currently selected context.
- **Delete**: Click to delete the currently selected section.
- **Arrange**: Click to open the "Arrange Sections" on page 473 dialog.
- **Go to**: Click to list the sections in the currently selected context and open one by clicking it.
- **Properties...**: Click to open the appropriate section properties: Email, Print or Web. See "Section properties dialogs" on page 468.
- **Includes...**: Click to open the "Section Includes" on page 473 dialog.
- **Finishing...** (Print Sections only): Click to open the Finishing tab in the "Print Section Properties" on page 469.
- **Sheet Configuration...** (Print Sections Only): Click to open the Sheet Configuration.
- **Master Pages**: Click to list the available Master Pages in the template, and open one by clicking it.
- **Master Page Properties**...: Click to open the currently selected Master Page's properties dialog; see "Master Pages" on page 365.

View Menu

- **50/75/100/150/200%**: Click to zoom the Workspace at the selected level.
- **Source View**: Click to show the HTML source for the template, including HTML Headers, CSS and HTML code.
- **Design View**: Click to show the template including all styles, text and images as well as the placeholders used for variable data.

- **Preview View**: Click to show the template as it will output with the current record, including all script executions, overflows, variable images, etc.
- **Show Edges**: Click to show or hide a colored border around elements on the page and the type of element that is highlighted.
- **Rulers**: Click to show or hide the rulers in the Workspace. Rulers only appear for Print contexts.
- **Margins and Guides**: Click to show or hide the margin lines and guides in the Workspace.
- **Snap to Guides**: Click to enable or disable snapping to guides and to margins when moving objects.
- **Virtual Stationery**: Click to enable or disable the visibility of the PDF Background image set in the Media.
- **Highlight Master Page Items**: Click to enable or disable a yellow border indicating an element in the section resides in a Master Page.
- **Object Resizing**: Click to enable or disable the ability to resize <div> elements on the page. See "Editing preferences" on page 459 for more fine-tuned control.

Window Menu

- **Show View**: Use the options in this menu to show or hide different panes of the UI.
  - **Properties**: Shows the Properties Pane
  - **Messages**: Shows the Messages Pane, see "Problems and messages" on page 493.
  - **Problems**: Shows the Problems Pane, see "Problems and messages" on page 493.
  - **Resources**: Shows the Resources Pane
  - **Outline**: Shows the Outline Pane
  - **Data Model**: Shows the Data Model Pane
  - **Scripts**: Shows the Scripts Pane
- **Reset Perspective**: Resets all toolbars and panes to the initial configuration of the module.
- **Preferences**: Click to open the software Preferences dialog.

Help Menu

- **Software Activation**: Displays the Software Activation dialog. See Activating your license.
- **Help Topics**: Click to open the PReS Connect 1.4.2 help system in the default web browser.

- **Contact Support**: Click to open the Objectif Lune Contact Page in the default system Web browser.
- **About PReS Connect Designer**: Displays the software's About dialog.
- **Welcome Screen**: Click to re-open 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 Properties 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 Section

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 as well as JavaScript expressions affecting single elements.
- **Class**: One or more classes that can be common to more than one elements. Used for CSS selections and JavaScript expressions that can affect multiple elements.

Other Section

These attributes are available depending on the item selected (in parenthesis).

- **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 (a paragraph not inside a table or a div) will function at whitespace elements.
- **Source (image)**: 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 PDF Page dialog.

- **Alternate text (image)**: The "Alt" text used when hovering over the image in a browser. Also used for accessibility.
- **Height (image)**: The specified height of the image. Defaults to the original image height in pixels.
- **Width (image)**: The specified width of the image. Defaults to the original image width in pixels.
- **Cell Spacing (table only)**: Defines the *cellspacing* attribute of the table which controls the spacing between cells in the table.
- **Cell Padding (table only)**: Defines the *cellpadding* attribute of the table which controls the padding inside each cell of the table.
- **Column Resizing (table only)**: Check to enable columns to be resized directly within the Workspace.
- **Detail Table (table only)**: Defines which detail table the repeat of the table is based on. The number of detail lines in the table is the number of the time the repeating row (see below) is repeated.
- **Title (table only)**: Defines the title of the table. This has no impact on the table's displays, only on accessibility of HTML pages and screen readers.
- **Repeat (table row not in <tfoot> or <thead> only)**: Defines if the row is affected by the detail table calculation. This row is the one repeated in a Dynamic Table.
- **Show Row (table row only)**: Use the drop-down to determine when the selected row appears when a dynamic table overflows. This option is only available in a row manually added inside of a Dynamic Table.

  - **Before page break**: The row will appear on all pages except the last one.
  - **At end of table**: The row will appear only on the last page.
  - **Always**: The row will appear on every page of the table.

- **Subtotal Line (table row inside a <tfoot> only)**: Defines the footer row as the place where the SubTotal is displayed. This is the row where a subtotal script is expected to display the result.

Geometry Section

These attributes are available for certain elements that have position or size attributes such as images and boxes.

- **X-Offset**: 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.
- **Y-Offset**: 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.
- **Height**: The height of the element, by default in pixels.

Page Section

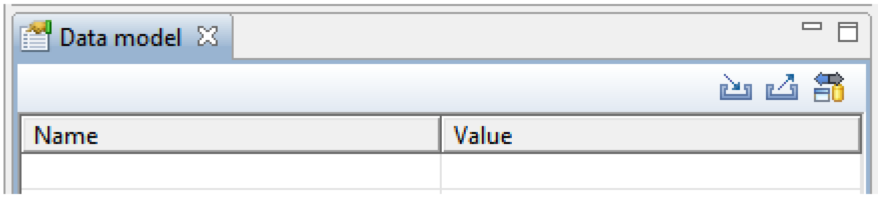
These attributes appear when selecting the *Page* node in the Outline Pane.

- **Master Page**: Which of the Master Pages to use for the template.

Data Model Pane

The Data Model Pane displays a Data Model used to help design the template, along with optional Extracted Data generally resulting from the execution of a Data Mapping Configuration. The information shown is the extracted information for the current record within the Record Set. It is also used as a navigation tool between records and all tables.

Data is displayed as a tree view, with the root level being the Record table, levels below it being detail tables, and any level below being called Nested Tables.



Pane Options

- **Minimize/Maximize**: Click to minimize or maximize the pane. See Moving and Merging Panes.
- **Import Data Model**: Click to browse to a Data Model File and import it. See "Importing a Data Model File" below.
- **Export Data Model**: Click to browse to a location to save the Data Model File and save it.
- **Synchronize Data Model**: Click to synchronize the data model with the one currently loaded in the open Data Mapping Configuration. Disabled if no configuration is currently open. See "Synchronizing Data Models" below.

Using the Data Model

When a Data Model is loaded inside of the Data Model Pane, it can be used to design templates by dragging the fields directly into the template; see "Variable Data" on page 333. If data is present (from a Data Model File or a Data Mapping Configuration), it is possible to preview the resulting data in the template using the Preview tab (see Workspace).

Importing a Data Model

There are three different ways to import a Data Model into a Template to help designing it.

Importing a Data Model File

Importing a Data Model File displays the file's data model structure into the Data Model Pane, with optional sample data for each field. To import a data model file, click the Import button at the top of the Data Model Pane.

Synchronizing Data Models

If a Data Mapping Configuration is open and contains fields in its record, it is possible to use the *Synchronize data models* button in the pane's toolbar to retrieve the model currently stored in the DataMapper Module's Data Model Pane.

Running a Data Mapping Configuration or Wizard

If executing a data mapping configuration or directly loading a data source, the resulting record set is loaded in the Data Model Pane. See "Loading data" on page 324.

Master Page Properties

The following properties are available for Master Page resources:

- **Name**: The name of the master page, displayed in all drop-downs where master page is called as well as in the "Resources Pane" on page 496.
- **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

The following properties are available for Media resources:

Properties Tab

- **Name**: The name of the media, displayed in all drop-downs where Media is called as well as in the Resources Pane.
- **Size** group: This group is read-only and only used to display the size selected in the linked Section Properties (see "Print Section Properties" on page 469), if the media is called by.
  - **Page Size**: The named page size.
  - **Width**: The width of the page.
  - **Height**: The height of the page.
  - **Orientation**: Whether the page is portrait or landscape.

Virtual Stationery Tab

If the output is sent to preprinted media, a copy of this media can be added here and will then appear as a background to the Template. This ensures that elements added to the template will correspond to their correct location on the preprinted media. When both a Virtual Stationery background PDF and a Master Page containing a PDF are used, they will both be added to the preview, the Master Page being "in front" of the Virtual Stationery PDF.

Media are not printed, unless you want them to; see "Printing virtual stationery" on page 372.

- **Front/Back group**: Defines the preprinted media used for the front and back of the Virtual Stationery.
  - **PDF**: Use the drop-down to select which PDF to display as a background for the page.
    - Use the **Browse** button to import a PDF, or choose None to ignore preprinted media on this page.
    - Use the **Page** button to open the PDF Page dialog to select which page of the PDF is shown as a background.
  - **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

This define the intended type of paper on which the Print Context is meant to be output.

- **Media Type**: The type of paper, such as *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 *Gloassy*, *High Gloss*, *Matte*, *Satin*, etc.
- **Back Coating**: The pre-process coating applied to the front 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**: Predefined hole pattern that specifies the pre-punched holes in the media, such as *R2-generic*, *R2m-MIB*, *R4i-US*, etc.

Problems and messages

Messages Pane

The **Messages** pane is shared between the **DataMapper** and **Designer** modules and displays any warnings and errors from the data mapping configuration or template.

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. This indicates the name of the step as defined in its step properties.
- **Date**: The date and time when the error occurred.

Problems Pane

The Problems pane displays any notifications or errors related to the template, its scripts, its code or output generation.

Log Filter

The log filter determines what kind of events are show in the Messages Pane.

- **Event Types** group:
  - **OK**: Uncheck to hide OK-level entries.
  - **Information**: Uncheck to hide information-level entries.
  - **Warning**: Uncheck to hide any warnings.
  - **Error**: Uncheck to hide any critical errors.
- **Limit visible events to**: Enter the maximum number of events to show in the Messages Pane. Default is 50.

Moving and merging panes

The PReS Connect interface for both the Designer and DataMapper module is highly configurable. Each panel in the application can be moved, with the exception of the Data Viewer and"Workspace" on page 517 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 theWorkspace, 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 options:
  - **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 Resource Pane displays the resources that affect the template and its output.

> **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 306 and "Print Options" on page 583) including page size and paper type. See "Media" on page 368 for more information.

Contextual menu

- **New Media**: Click to create a new media and open its properties.
- **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.

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 365 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 365 for more information.

Contexts

Contexts hold the actual content of the template that is used to generate output. See "Contexts" on page 218 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 Web Page Context**: Click to create a new Web Page 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 218 for more information.

Sections

Sections hold part of the contents within a specific context. See "Sections" on page 389 for more information.

Contextual menu

- **Set as Default** (Email and Web 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, Print or Web. See "Section properties dialogs" on page 468.
- **Includes...**: Click to open the "Section Includes" on page 473 dialog.
- **Finishing...** (Print Sections only): Click to open the Finishing tab in the "Print Section Properties" on page 469
- **Sheet Configuration...** (Print Sections Only): Click to open the Sheet Configuration dialog; see "Master Pages" on page 365 and "Media" on page 368.

Images

Images are graphical elements that can be added to the page for display, either statically or dynamically. See "Images" on page 276 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. 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.

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. Currently, fonts must be set through the CSS Stylesheet and do not appear in the fonts drop-down menu.

Currently supported font types: otf, woff, ttf, svg. Fonts must be set to *installable* to be useable in the output.

Please see the Tips & Tricks post for details on how to embed the fonts.

JavaScripts

JavaScripts are scripted programs that can run on either Web output (they are added to the page header) or in Print Output (they are executed on the page after it has run, before the actual output is created in the selected format). See "Using JavaScript" on page 447 for more information.

Contextual menu

- **New Javascript**: Click to create a new JavaScript resource.
- **New Remote Javascript**: Click to add a Remote JavaScript resource. See "Using JavaScript" on page 447 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.
- **Delete**: Click to delete the resource. This is the same as pressing the Delete key while the resource is selected.

Stylesheets

Stylesheets control how contents appears on the page. It defines spacing, color, size and other properties of elements on the page. See "Styling templates with CSS files" on page 396 for more information.

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 396 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.
- **Delete**: Click to delete the resource. This is the same as pressing the Delete key while the resource is selected.

Snippets

Snippets are pieces of HTML or JSON code that can be inserted within sections and master pages, dynamically or statically. See Snippets for more information.

Contextual menu

- **New HTML Snippet**: Click to create a new HTML Snippet resource.
- **New JSON Snippet**: Click to create a new JSON Snippet resource.
- **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.

Scripts Pane

The Scripts pane contains all of the scripts that are used to replace data on your page, or to modify its look. Scripts can be exported and imported via the buttons or through drag & drop between the Scripts Pane and any location on the computer.

> **Technical**
>
> Scripts included here are completely distinct from the JavaScript resources found in the Resources Pane. 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 PReS Connect JavaScript API (see "API" on page 171), such as the Record. JavaScript resources, on the other hand, are only executed in a client *after* the content creation is done, which is generally in the browser.

> **Note**
>
> Scripts are always executed top-to-bottom. They can be dragged higher or lower in the pane to change their order of execution. For example, content loading scripts (loading snippets, for instance) must be present before scripts that replace data within that loaded contents.

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:
  - **Script**: Adds a new empty basic script
  - Text Script: The default script that is created when a Record Field is dragged on to the page.
  - Dynamic Image Script: This script can be added and, providing it refers to an image by selector, can dynamically change the image for each record.
  - Email To Script: This script is automatically added on any new Email template or Context and defines the email address where each record output is sent.
  - Email Subject Script: This script can be added to Email contexts and defines the subject of the email that is sent.
  - Conditional Content Script: This script can be added to conditionally show or hide any element in the template.
  - **Folder**: Adds a folder in which scripts can be placed for easier management.
- **Collapse All**: Collapses all the folders, hiding the scripts inside of them.

Contextual menu options

- **Duplicate**: Click to create an exact copy of the script.
- **Delete**: Click to delete the selected script. This does not delete the element or text the script refers to.
- **Rename**: Click to open a rename dialog to rename the script. This is the same as changing the **Name** field in the script's properties.
- **Enable/Disable**: Click to trigger the script to be enabled or disabled. Disabled scripts are greyed out and italic and will not be executed.

> **Note**
>
> Please see Script Generators for more information on the different Wizards that are available.

Scripts Pane column

- **Name**: The friendly name added to better identify the script.
- **Selector**: Displays the initial text, or javascript selector, that the script applies to.

Note

Fields from the Data Model Pane can be dragged directly into the Scripts pane to create a Text Script. This is useful if, for instance, when using snippets that already contain placeholders for text scripts. Additionally, Text scripts can be dragged into any section to add the script's placeholder at the insert location.

Script Properties

Scripts are used to modify the contents of the template, as its output is produced.

Here are the properties of a regular Script:

- **Name**: The name of the script, making it easier to identify it.
- **Find**: 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.
- **Expression**: This box contains the script itself.
- **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.
- **Apply**: Saves changes made to the script and applies the changes in the template without closing the dialog.

In PReS Connect Designer, scripts are built using the Connect JavaScript API. For more information, see "API" on page 171.

Script Folder Properties

The Script Folder Properties dialog controls when scripts are executed in the output, as well as the name of the script folder.

This dialog has the following properties:

- **Name**: Enter a name for the folder to rename it.
- **Execution scope**: Uncheck any of the entries in this list to prevent all scripts within the folder to execute in that entry. For example, if the "Print" context is unchecked, the scripts in the folder will not run when generating Print output.

> **Technical**
>
> Disabling script execution in certain contexts or sections helps with performance, since scripts normally run in your template whether or not their placeholder or selector is present. It is highly recommended to disable any script not relevant to specific sections or contexts.

Control Script

A Control Script is a script that runs before pages and variable data are executed are can be used to control how different sections of the context are handled. Control Scripts do not currently operate through Wizards, rather they are scripted directly and rely on the Control Script API. New Control Scripts added to the template will contain an example script showing a few samples.

To learn more about the Control Script API, please see Control Script API.

Edit Script dialog

Script wizards are simplified interfaces for common scripts in Templates.

These are the different Script wizards that are available.:

- Text Script: The default script that is created when a Record Field is dragged on to the page.
- Dynamic Image Script: This script can be added and, providing it refers to an image by selector, can dynamically change the image for each record.
- Email To Script: This script is automatically added on any new Email template or Context and defines the email address where each record output is sent.
- Email Subject Script: This script can be added to Email contexts and defines the subject of the email that is sent.
- QR Code Script: This script controls the contents of a "QR Code" on page 256.
- **Pie Chart Script**: This script controls the contents of Pie Chart Objects.

Here are the options visible in Script wizards:

- **Name**: The name of the script, making it easier to identify it.
- **Find**: 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.
- **Wizard Results**: Displays a list of the data that is sent to replace 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. Note that only fields from the Record are displayed, not from any detail tables.
  - **Format**: A special formatting modifier applied to the Field; see "Formatting variable data" on page 338.
  - **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.
- **Options** (only 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.
- **Apply**: Saves changes made to the script and applies the changes in the template without closing the dialog.

Bar Chart script dialog

[TBD]

Barcode script dialog

Here are the options visible in the Barcode Script Generator:

- **Name**: The name of the script, making it easier to identify it.
- **Find**: 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.
- **General Tab**:
    - **Type**: Which type of barcode to display. The options below depend on which type of barcode is selected.
    - **Value**: The data to display in the barcode. Currently no check is being done to verify that the barcode values are correct for the type.
    - **Options**:
        - **KIX**:
            - **Show human readable text**: Check to have human-readable (alphanumerical) characters appear underneath the barcode for verification of its value.

    - **QR/QR Micro**:
        - **Error Correction**: Select the level of error correction to apply. The size of the barcode depends on the error correction - higher correct means a larger barcode.
        - **Encoding**: Which encoding (and thus data) that is accepted in the barcode:
            - **Numeric**: Numeric characters up to a maximum of 7089 digits.
            - **Alphanumeric**: Numeric and alphanumeric characters. Limited to uppercase letters, spaces and the following characters: dollar ($), percent (%), star (*), plus (+), minus (-), period (.), slash (/), colon (:), semicolon (;), at (@) and comma (,) up to a maximum of 4296 characters. Note that lowercase characters are automatically converted to uppercase.
            - **Kanji**: All double-byte characters.
            - **Byte**: [TBD]
            - **Raw**: [TBD]
        - **Symbol size**: [TBD]
        - **Parse Text**: [TBD]
    - **Royal Mail 4 State (RM4SCC)**
        - **Show human readable text**: Check to have human-readable (alphanumerical) characters appear underneath the barcode for verification of its value.
        - **Show check digit**: When human readable text appears, also show the check digit that is added automatically at the end of the string.
- **Appearance Tab**:
    - **Width**: [TBD]
    - **Height**: [TBD]

Conditional script dialog

Conditional script generators can show or hide elements on the page depending on certain conditions and values. Currently they can only be added by right-clicking any element on the page and clicking "Make Conditional". If the current element does not have an ID, one will be automatically generated.

Though an ID is added automatically, the conditional script can function with a class selector. It cannot, however, affect a Text selection.

Conditional Script Generator Properties:

- **Name**: The name of the script, making it easier to identify it.
- **Find**: 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.
- **Action**: Use the drop-down to select whether to **Show** or **Hide** the element when the condition below is true.
- **Data Field**: Use the drop-down to select which data field in the record the condition will be based on.
- **Condition**: Select which kind of condition is checked. Possible options are: **Equal to**, **Not equal to**, **Contains**, **Does not contain**, **Begins with**, **Ends with**.
- **Value**: The value used for the conditional check.

For example, you could check whether the Data Field "Gender" is "Equal To" the value "Mr", in order to show a paragraph or an image applying only to male customers.

Dynamic Image script dialog

The Dynamic Image Script Generator is used to dynamically modify the source of an image element in the template for each content item output by the template. The result of the script Wizard must be a valid path to an image resource (see "Resources" on page 439 for more information).

Here are the options visible in the Dynamic Image Script Wizard :

- **Name**: The name of the script, making it easier to identify it.
- **Find**: 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.
- **Wizard Results**: Displays a list of the data that is sent to replace the result of the script:
  - **Prefix**: Static text to use before the set field. For Dynamic Image scripts, the default prefix is **images/**.

- **Field**: A drop-down to select which field contents to use in the script. Note that only fields from the *Record*tab are displayed, not from any detail tables.
    - **Format**: A special formatting modifier applied to the Field. The available formatting functions depend on the field type in the data model. For example, text fields can be set to uppercase or lowercase. Date fields can be set to long and short dates or different time displays.
    - **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.
- **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.
- **Apply**: Saves changes made to the script and applies the changes in the template without closing the dialog.

Email BCC script dialog

The Email BCC Script Wizard defines an additional email address to which the email will be sent for this record. The result of the Wizard must be a valid email address and will be used in the "BCC" field of the email.

Here are the options visible in the Email BCC Script Wizard:

- **Name**: The name of the script, making it easier to identify it.
- **Find**: The Selector or Text to apply the result of the script to. This option is read-only for this script.
- **Wizard Results**: Displays a list of the data that is sent to replace the result of the script:
    - **Prefix**: Static text to use before the set field.
    - **Field**: A drop-down to select which field contents to use in the script. This field should contain an email address. Note that only fields from the *Record*tab are displayed, not from any detail tables.

  - **Format**: A special formatting modifier applied to the Field. The available formatting functions depend on the field type in the data model. For example, text fields can be set to uppercase or lowercase. Date fields can be set to long and short dates or different time displays.
  - **Suffix**: Static text to use after the set field.
- **[+]**: 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.
- **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.
- **Apply**: Saves changes made to the script and applies the changes in the template without closing the dialog.

Email CC script dialog

The Email CC Script Wizard defines an additional email address to which the email will be sent for this record. The result of the Wizard must be a valid email address and will be used in the "CC" field of the email.

Here are the options visible in the Email CC Script Wizard:

- **Name**: The name of the script, making it easier to identify it.
- **Find**: The Selector or Text to apply the result of the script to. This option is read-only for this script.
- **Wizard Results**: Displays a list of the data that is sent to replace the result of the script:
  - **Prefix**: Static text to use before the set field.
  - **Field**: A drop-down to select which field contents to use in the script. This field should contain an email address. Note that only fields from the *Record*tab are displayed, not from any detail tables.
  - **Format**: A special formatting modifier applied to the Field. The available formatting functions depend on the field type in the data model. For example, text fields can be

    set to uppercase or lowercase. Date fields can be set to long and short dates or different time displays.
  - **Suffix**: Static text to use after the set field.

- **[+]**: 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.
- **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.
- **Apply**: Saves changes made to the script and applies the changes in the template without closing the dialog.

Email From script dialog

The From Script Wizard overwrites the default "From" Address in the Send Email dialog with the result of the script below. The result should be a valid, fully-formed email address.

Here are the options visible in the Email From Script Wizard:

- **Name**: The name of the script, making it easier to identify it.
- **Find**: The Selector or Text to apply the result of the script to. This option is read-only for this script.
- **Wizard Results**: Displays a list of the data that is sent to replace the result of the script:
  - **Prefix**: Static text to use before the set field.
  - **Field**: A drop-down to select which field contents to use in the script. This field should contain an email address. Note that only fields from the *Record*tab are displayed, not from any detail tables.
  - **Format**: A special formatting modifier applied to the Field. The available formatting functions depend on the field type in the data model. For example, text fields can be set to uppercase or lowercase. Date fields can be set to long and short dates or different time displays.
  - **Suffix**: Static text to use after the set field.

- **[+]**: 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.
- **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.
- **Apply**: Saves changes made to the script and applies the changes in the template without closing the dialog.

Email PDF Password script dialog

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. The result of the script will be the password necessary to open the PDF when it is received by email.

Here are the options visible in the Email PDF Password Script Wizard:

- **Name**: The name of the script, making it easier to identify it.
- **Find**: The Selector or Text to apply the result of the script to. This option is read-only for this script.
- **Wizard Results**: Displays a list of the data that is sent to replace the result of the script:
  - **Prefix**: Static text to use before the set field.
  - **Field**: A drop-down to select which field contents to use in the script. Note that only fields from the *Record*tab are displayed, not from any detail tables.
  - **Format**: A special formatting modifier applied to the Field. The available formatting functions depend on the field type in the data model. For example, text fields can be set to uppercase or lowercase. Date fields can be set to long and short dates or different time displays.
  - **Suffix**: Static text to use after the set field.
- **[+]**: 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.
- **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.
- **Apply**: Saves changes made to the script and applies the changes in the template without closing the dialog.

Email Reply-To script dialog

The Reply-To address is often used when sending email campaigns and to do tracking of email replies. This script Wizard modified the reply-to address in the email. The result of this script should be a valid, fully-formed email address.

Here are the options visible in the Email Reply-To Script Generator:

- **Name**: The name of the script, making it easier to identify it.
- **Find**: The Selector or Text to apply the result of the script to. This option is read-only for this script.
- **Wizard Results**: Displays a list of the data that is sent to replace the result of the script:
  - **Prefix**: Static text to use before the set field.
  - **Field**: A drop-down to select which field contents to use in the script. This field should contain an email address. Note that only fields from the *Record*tab are displayed, not from any detail tables.
  - **Format**: A special formatting modifier applied to the Field. The available formatting functions depend on the field type in the data model. For example, text fields can be set to uppercase or lowercase. Date fields can be set to long and short dates or different time displays.
  - **Suffix**: Static text to use after the set field.
- **[+]**: 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.

- **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.
- **Apply**: Saves changes made to the script and applies the changes in the template without closing the dialog.

Email Subject script dialog

The Email Subject Script Wizard defines the subject line of emails that are sent in Email Contexts. The result will appear in the email as the Subject line.

Here are the options visible in the Email Subject Script Generator:

- **Name**: The name of the script, making it easier to identify it.
- **Find**: The Selector or Text to apply the result of the script to. This option is read-only for this script.
- **Wizard Results**: Displays a list of the data that is sent to replace the result of the script:
  - **Prefix**: Static text to use before the set field.
  - **Field**: A drop-down to select which field contents to use in the script. Note that only fields from the *Record*tab are displayed, not from any detail tables.
  - **Format**: A special formatting modifier applied to the Field. The available formatting functions depend on the field type in the data model. For example, text fields can be set to uppercase or lowercase. Date fields can be set to long and short dates or different time displays.
  - **Suffix**: Static text to use after the set field.
- **[+]**: 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.
- **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.

- **Apply**: Saves changes made to the script and applies the changes in the template without closing the dialog.

Email To script dialog

The Email To Script Wizard defines the email address to which the email will be sent for this record. The result of the Wizard must be a valid email address and will be used in the "To" field of the email.

Here are the options visible in the Email To Script Generator:

- **Name**: The name of the script, making it easier to identify it.
- **Find**: The Selector or Text to apply the result of the script to. This option is read-only for this script.
- **Wizard Results**: Displays a list of the data that is sent to replace the result of the script:
  - **Prefix**: Static text to use before the set field.
  - **Field**: A drop-down to select which field contents to use in the script. This field should contain an email address. Note that only fields from the *Record*tab are displayed, not from any detail tables.
  - **Format**: A special formatting modifier applied to the Field. The available formatting functions depend on the field type in the data model. For example, text fields can be set to uppercase or lowercase. Date fields can be set to long and short dates or different time displays.
  - **Suffix**: Static text to use after the set field.
- **[+]**: 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.
- **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.
- **Apply**: Saves changes made to the script and applies the changes in the template without closing the dialog.

Line Chart script dialog

[TBD]

Pie Chart script dialog

Here are the options visible in Chart Script wizard:

- **Name**: The name of the script, making it easier to identify it.
- **Find**: 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.
- **Input Type**: Use the drop-down to select the source of the data to add to the Chart. The selection changes the options below:
  - **Static Labels**: Select to use a static number of data lines below. The chart will always have the same number of items.
    - **Data List**: Lists the data lines that are part of the Chart. Each line represents a section of the pie as well as a label if they are shown.
      - **Labels**: The text of the label to display next to the Chart or within the legends.
      - **Values**: The value that will be used to create the Chart. This is the name of a field within the Data Model.
    - **Add**: Click to add an entry to the Data List. Opens the Edit Label Properties dialog.
    - **Delete**: Click to delete the currently selected line in the Data List.
    - **Move Up**: Click to move the currently selected line up one position.
    - **Move Down**: Click to move the currently selected line down one position.
  - **Dynamic Labels**: Select to use data from a detail table to fill the Chart dynamically. At least one detail table must be available in the Data Model Pane for this option to be functional.
    - **Details**: Use the drop-down to select which detail table provides the data for the Chart.
    - **Labels**: Use the drop-down to select which field within the detail table contains the text for the labels shown in the Chart.
    - **Values**: Use the drop-down to select which field within the detail table contains the numerical values used to build the Chart.

Text script dialog

The Text Script Wizard is used first and foremost for the replacement of placeholders on the page, and is created automatically when dragging a data field or a detail table on the page. It also appears when adding a QR Code in order to determine the data added to that barcode. The result of this script is either the text appearing on the page, or the QR Code data.

Here are the options visible in the Text Script Wizard:

- **Name**: The name of the script, making it easier to identify it.
- **Find**: 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.
- **Wizard Results**: Displays a list of the data that is sent to replace the result of the script:
    - **Prefix**: Static text to use before the set field.
    - **Field**: A drop-down to select which field contents to use in the script. Note that only fields from the *Record*tab are displayed, not from any detail tables.
    - **Format**: A special formatting modifier applied to the Field. The available formatting functions depend on the field type in the data model. For example, text fields can be set to uppercase or lowercase. Date fields can be set to long and short dates or different time displays.
    - **Suffix**: Static text to use after the set field.
- **[+]**: 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.
- **Options group**:
    - **Insert method**: Defines where and how the script inserts its results:
        - **HTML**: Inserts the result as processed HTML. If the result includes HTML tags and elements, they are inserted into the source code, and displayed as their result. For instance, `<b>this is bold</b>` will be displayed as **this is bold**.

    - **Text**: Inserts the result as-is, meaning HTML tags and elements will be displayed in the output.
    - **Attribute**: Inserts the result into the specified attribute. For example if the attribute is `href` and the script refers to a hyperlink (<a> element), the script will modify the link's destination.
- **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 to a regular script, showing the code behind the Wizard. Note that this action is not reversible.
- **Apply**: Saves changes made to the script and applies the changes in the template without closing the dialog.

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 395).

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 402 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 "Determining the order in which style sheets are read" on page 404.

Workspace

The Workspace pane is where everything comes together. It is the contents of the page, the WYSIWYG editor that shows how 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.

Source Tab

The source tab displays the HTML source for the template, 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.

To the left of the Source tab, a bar helps visually identify the start and stop of an element. For example if clicking on the opening <table> element, this bar highlights the whole <table> and all its contents, until the ending </table> tag.

The top of the Design tab contains an area with the following options:

- **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. If an element is selected in the breadcrumbs and the Backspace key is pressed, that element is deleted.
- **Context Selector**: Displays the current context and drops down to list 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.

Design Tab

The design tab show 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.

The top of the Design tab contains an area with the following options:

- **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. If an element is selected in the breadcrumbs and the Backspace key is pressed, that element is deleted.
- **Context Selector**: Displays the current context and drops down to list 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.
- **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.
- **Responsive Design View**: Use the drop-down to select a specific screen width, to test the design for different devices. Easier than resizing the window, right? Not available in Print contexts.

Preview Tab

The preview tab shows the template as it will output with the current record, including all script executions, overflows, variable images, etc. It is still possible to edit the template while in Preview mode - however, editing around placeholders and within dynamic tables may be a little harder. It is recommended to do editing in the Design mode.

Web Tab (HTML context only)

The Web tab shows the result of the template as rendered by the Gecko rendering engine. It is a good indication of how an HTML template would display in a visitor's browser, especially if they are using FireFox (which uses the Gecko engine).

Toolbars

In the Designer module, the following buttons are available in the top toolbar:

- **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 583 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.
    - **Send Test Email**: Opens the Send Test Email dialog.

- **External Browser**: Opens the current template's Preview in the system default browser. Useful for testing scripts and HTML output.
- **Send COTG Test**: Click to open the Send COTG Test dialog, to send the current Web Context to the Capture OnTheGo Application. See this how-to: Testing a COTG template.

- **Forms**
  - **Insert Form**: Inserts a <form> element.
  - **Insert Fieldset**: Insert a <fieldset> element.
  - **Insert Text Field**: Inserts a <input type="text" /> element. A drop-down is available to insert other fields, such as a URL, Password etc.
  - **Insert Text Area Field**: Inserts a <textarea> element.
  - **Insert Label**: Inserts a <label> element.
  - **Insert Checkbox**: Inserts a <input type="checkbox"> element.
  - **Insert Radio Button**: Inserts a <input type="radio"> element.
  - **Insert Select Field**: Inserts a <select> element and add multiple possible options to it.
  - **Insert Button**: Inserts a <button type="submit"> element at the current cursor location.

  For information about Forms and Form elements, see "Forms" on page 266 and "Form Elements" on page 269.
- **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 407.
  - **Insert Vertical Guide**: Click to insert a new horizontal guide; see "How to position elements" on page 407.
- **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.

- **Form Wizard**
    - **Form Wizard**: Click to open the Form Wizard to add a form to a Web Context. See "Forms" on page 266 and "Form Elements" on page 269.
    - **Validation Settings**: Click to open the Validation settings dialog to change the validation settings on the currently selecting tools. See "Forms" on page 266.
- **Table Manipulation**
    - **Insert Standard Table...**: Inserts a table with a specific number of columns and rows through the "Table" on page 280 Wizard.
    - **Insert Dynamic Table...**: Inserts a dynamic table where the number of rows is determined by a Details table, through the "Dynamic table" on page 344 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 276.
  - **Insert Image from Address...**: Inserts an Image using a URL instead of a resource, at the current location of the pointer and opens its properties. See "Images" on page 276.
  - **Insert Barcode**: Displays a list of available barcodes. Click on one to insert it on the page. The Text Wizard is opened for the barcode data.
  - **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 274.
  - **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.
- **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 Decoration**
    - **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~~.
- **Indentation**
    - **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.

- **Position**
  - **Superscript**: Makes the currently selected text a $^{\text{superscript}}$.
  - **Subscript**: Makes the currently selected text a $_{\text{subscript}}$.
- **Remove Formatting**: Remove any and all styles, text decorations and other formatting from the selected text. Indentation is not affected.
- **Welcome Screen**: Click to re-open the Welcome Screen.

Welcome Screen

The **Welcome Screen** appears when first starting up PReS Connect. It offers some useful shortcuts to resources and to recent documents and data mapping configurations.

The Welcome Screen can be brought back in two ways:

- The **Welcome Screen** button in the "Toolbars" on page 519.
- From the Menus in **Help**, **Welcome Screen**.

Contents

- **Activation**: Click to open the **Objectif Lune Web Activation Manager**.
- **Release Notes**: Opens the current **Release Notes** for PReS Connect.
- **Website**: Opens the PReS Connect website.
- **Take A Tour**: Click to open the YouTube Playlist giving you a tour of the software.
- **Use the DataMapper to...**:
  - **Create a New Configuration**: Opens the Creating a New Configuration screen.
  - **Open an Existing Configuration**: Click to open the standard **Browse** dialog to open an existing data mapping configuration.
  - **Recent Configurations**: Lists recently used configurations. Click any configuration to open it in the DataMapper module.
- **Use the Designer to...**:
  - **Create a New Template**: Lets you choose a Context to create a new template without a Wizard.
  - **Browse Template Wizards**: Displays a list of available Template Wizards, producing premade templates with existing demo content; see "Creating a template" on page 420.
  - **Open an Existing Template**: Click to open the standard **Browse** dialog to open an existing template.

    - **Recent Templates**: Lists recently used templates. Click any template to open it in the Designer module.
- **Other Resources**:
    - **Documentation**: Opens this documentation.
    - **Courses (OL Learn)**: Opens the Objectif Lune e-Learning Center.
    - **User Forums**: Opens the Questions & Answer forums.

Print Using Standard Print Output Settings

When using the **File > 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**: Use the drop-down to select existing Output Creation Presets. Use the Gear button to edit the currently selected Preset or to reload the list of Presets from the system. ⚙
    - **Job Creation**: Use the drop-down to select existing Job Creation Presets. 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.

    > **Note**
    >
    > The Default output type of 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 specific 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 579 and "Sorting Options" on page 580 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.
- **Options** Group:
    - **Use grouping, splitting and slip sheets**: Check to use grouping options as defined in the currently selected Job Creation preset.
- **Advanced**: Click to open the "Print Using Advanced Printer Wizard " belowwhere 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**: Click to produce print output according to the current settings.

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 583 page.

Print Options

This section describes the Print Options, which is the first page of the **Advanced Print Wizard** as well as the Output Creation Settings Preset .

This page is the most important of the Advanced Print Wizard.
The pages shown throughout the Wizard are determined by the selections made on this page.

The choices can be broken down as follows:

- **Printer** section:
  - **Model**: Use the drop-down to select the printer language / output type that will be generated.
    Connect output options cover a range of industry standard print output types. These include AFP, IPDS, PCL, PDF and PostScript (including PPML, VIPP and VPS variants), with a range of quality settings available.

    > **Note**
    >
    > By default, Connect displays only the PDF output option, but other print output types can be added to the Printer Model drop down list via the Gear button ⚙

    For more information on how to do this, see "Adding print output types to the Print Wizard" on page 308.
- **Output Options** section:
  - **Output Local** checkbox: 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 filename.

- **Directory**: Select to output to a local folder on the machine.
  - **Job Output Mask**: The name of the file that will output.
    You can use ${template} as a variable for the name of the Designer Template used to generate the output.
  - **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.
- **Windows Printer**: Select to send the Print Job to a Printer Queue. The job is rendered as a PDF before being printed through the Windows driver.
  - **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 (PDF output only)**:
  - **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**:

        - **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 resized.
        - **Shrink to printable area**: Select to shrink any page to fit the page area. Pages smaller than the paper size are not resized.
- **Production Options:**
    - **Booklet Imposition** checkbox: Check to tell the printer to generate a booklet for the print output. Booklet options are set in the "Booklet Options" on page 588 page. This option is unselected by default unless selected in the Designer "Print Section Properties" on page 469.
    - **Cut and Stack Imposition** checkbox: Check to enable Cut & Stack Imposition, which is set in the "Imposition Options" on page 589 page.
    - **Add Inserter marks** checkbox: Check to enable inserter mark functionality, which is set in the "Inserter Options" on page 591 page.
    - **Override Finishing options** checkbox: Check to configure custom "Finishing Options" on page 578, such as binding.
    - **Print virtual stationery** checkbox: Check to enable virtual stationery in the output.
    - **Use grouping** checkbox: Check to configure grouping of output into jobs, job segments or document sets. See "Grouping Options" on page 581.
    - **Include meta data** checkbox: Check to add meta data to the output. This can be done at Job, Job Segment, Document, Document Set and Page level. See "Metadata Options " on page 582.
    - **Separation**: Check to activate the "Separation Options" on page 587 page of the wizard.
    - **Add additional content** checkbox: Check to activate the "Additional Text" on page 540 page of the wizard.
- **Records** section:
    - **Record Range**: Allows selection of a range of records or a custom selection.
      You can specific 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.
- **Copies** section:
    - **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** section:
  This section is valid for AFP, IPDS and PCL only. It applies to elements within the record that are shades of gray, rather than black or white.
  - **Black Threshold Percentage**: The percentage of shading at which the element will appear as full black, rather than dark gray.
  - **White Threshold Percentage**: The percentage at which the element will appear as full white, rather than light gray.

Printer Settings

The Printer Settings page defines options on the printer. It is available for PostScript (and the VIPP and VPS variants of PostScript) only.

- **Use Media Position** checkbox: Enables the *Position* column and removes the *Weight*, *Type* and *Color* columns, to simplify the selection of media, using only the Media Position option.

- Tray selection columns:
  - **Media**: Lists the medias 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.
  - **Weight**: Enter a weight for the paper.
  - **Type**: Use the drop-down to select which type of stock to use on the printer.
  - **Color**: Use the drop-down to select which color the paper should be on the printer.

Booklet Options

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** sections 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.

Imposition Options

The imposition options page sets the repetition, order, margins and marks for imposition in the output.

- **Sheet Size** group:
  - **Final Media Size**: Use the drop-down to select the size of the media where the output is printed. The size of the media should be equivalent to the initial section size multiplied by the number of repetitions, added with the margins and spaces

between the repetitions.
If *Custom* media size is selected, enter the custom **Width** and **Height** values.

> **Note**
>
> The Sheet Size cannot be altered if a Cover Page was selected in the "Booklet Options" on page 588 Page.

- **Sheet Rotations**: Select aspect ratio of media (Landscape or Portrait), or allow Connect to automatically determine the proper aspect ratio.
- **Scale to fit**: Modify the size of each repetition to fit within the final media size.
- **Rotate final output Sheet 180 degrees (upside down)**: Select to flip the output upside down.

- **Repetition** group:
Allows selection of how many sections are to be placed, both Horizontally and Vertically. This is the total number of items, not the number of additional items being placed.

> **Note**
>
> If *Booklet Binding* were selected, some of these settings will be determined by the options made within the "Booklet Options" on page 588 Page and they cannot be altered here.

- **Space Between** group:
Allows selection of the amount of blank space to add between each repetition.
- **Order** group:

> **Note**
>
> If *Booklet Binding* were selected, some of these settings will be determined by the options made within the "Booklet Options" on page 588 Page and they cannot be altered here.

    - **Page Order**: Select in which direction to go when adding sections to the output:
        - **Left to right, then top to bottom**
        - **Right to left, then top to bottom**
        - **Top to bottom, then left to right**
        - **Top to bottom, then right to left**
    - **Stack Depth**: Enter a stack depth or use the arrows to increment or decrement.
    - **Reverse Pages**: Select this option to reverse the order of pages.
      This would print the final record on the first page and the first record on the last page.
    - **Force simplex**: Select this option to make the output Simplex, rather than the imposition default of Duplex.
- **Bleed Margins** group:
    - **Top, Bottom, Left, Right**: Enter the bleed margins for each side of the page.
- **CropMarks** group:
    - **Type**: Use the drop-down to select the type of crop marks to add to the page.
    - **Offset**: How much separation (if any) to leave between the vertical and horizontal corner markings.
    - **Width**: Select the width of the crop mark lines.
    - **Length**: Select the Length of the crop mark lines.

Inserter Options

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 page influence the position of the markings set on the next page: **"Mark Position Options" on page 592**.

- **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 592 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.
    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".

- **Clear Background Area**: Check 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 a measure for the margins on each side of the OMR Marks.

- **Custom OMR mark sizing**: If supported by the chosen Mark Configuration you can select a Custom OMR size.
  You can 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.

Mark Position Options

This 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 591 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 sections 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%
- 75% ⌄ **Zoom Level**: Use the drop-down to select a predefined level or enter a zooming percentage.

Finishing Options

Use this dialog to force specific finishing options, instead of using finishing options that were set in the Template's Media and Section options.
This is only applied when producing print output. It does not modify the original finishing options in either the Section or the Template.

- **Ignore section level finishing**: Check to override finishing options at the document level only.
- **Section to edit**: Use the drop-down to select which Section to apply the options below. The Document level is also listed to edit document-level finishing.
  - **Settings:** Click the settings button ⚙ to bring up control options:
    - **Reload:** Restores the current section's properties to the default values set in the template for this section
    - **Reload All:** Restores all section properties to the default values set in the template for each section.
    - **Apply finishing from:** Displays a list of available sections. Clicking on a section name loads that section's properties into the current section to edit.
    - **Apply current finishing to all sections:** Applies the current properties to all sections.
- **Binding** group:
  - **Style**: What type of Binding to request on the printer. This includes Stapled, Glued, Stitched, Ring, and various other options..
  - **Side**: Sets the side of the paper that the Binding is to occur.
  - **Location**: Sets where the binding is to occur, if applicable.
    The selections available here are 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 actually available to you 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.
    - **Style** The style of hole punches for the selected Binding option.
    - **Pattern Catalog ID**: The Catalog ID of the selected Binding option.

Grouping Options

The Grouping options separates the job output into multiple blocks that can then be physically separated using split sheets in the printer.

- **Grouping Tabs**: Jobs can be grouped at three different levels, each of which is each contained in a tab in this section. The groups/tabs are:
    - **Job** Grouping Fields
    - **Job Segment** Grouping Fields
    - **Document Set** Grouping Fields

  All the Fields available to be used for Grouping are contained within the *Available Fields* box in each tab. Fields that you want to use for Grouping need to be added to the *Selected Fields* box via the arrows between the two boxes. Simply select the Field(s) you want to move and click the 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.

- **Page Break Grouping**: Check to enable page break grouping, which separates different groups by the number of pages they contain. For example, enabling the Document Set Grouping Level and creating a page range from 1-5 and 6 to Largest, will create two groups. The first will contain all document sets of 1 to 5 pages, the second will contain any document set of 6 or more pages.
    - **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.
    - **Grouping list**: 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".
  - **Generate page break ranges in reverse order**: Reverses the order of the groups created. By default, grouping will be from smallest to largest. Checking this option creates groups from largest to smallest.
  - **Generate page break range groups after normal grouping**: Check this option to first group using the levels above, following which page break grouping are applied. This creates two different levels of grouping, applied in order.

Metadata Options

The Metadata Option page defines metadata tags that will be added to the output file when producing PDF and AFP output in the Output Creation Presets. Metadata tags are ignored in all other output types. The tags are added to each of the levels, as indicated by the tabs on top: **Job**, **Job Segment**, **Document**, **Document Set**, and **Page** Tags.

In each of these levels, a list of tags is available:

- **Always create meta data for this level even when fields are selected**: Select 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**: Name of the metadata 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**: Displays the type of field being used - either Text or Data Field.
- **Source**: For Data Fields only. The Field name from the data mapping configuration whose value will be used for this tag.
- ➕ **Add Field**: Click to add a new tag to the current level. The Field Selection dialog appears. Select either **Add field meta data** or **Add text meta data**.
  When adding field meta data select a field name from the Field List and click OK to add it as a tag of the same name.
- ➖ **Delete Field**: Click to delete the currently selected tag.
- ⬆ **Move Up**: Click to move the currently selected tag one position up.
- ⬇ **Move Down**: Click to move the currently selected tag one position down.

Separation Options

This page defines how to separate the jobs using subsets, slip sheets, or jogging.

- **Sheet Count Splitting** group.
  This group allows for the splitting of output based upon a pre-determined number of pages
  - **Split**: Use the drop-down to select how to split.
    - **None**: Select to ignore sheet count splitting entirely.
    - **At exactly**: Select to create a split at a specific sheet number.
  - **Every**: Enter the number of sheets at which to split the output.
- **Separation Settings** group.
  This setting is only available if no Sheet Count Split were 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.

Additional Content

There are four different types of additional content that can be added at print time. **Text**, **Images**, **Barcodes** and **OMR Marks**. They are used to add static or variable content when generating output. This is useful when driving custom processes on machines using either Barcodes or OMR Marks, as well as allowing the addition of last minute information through text and images.

Additional Text

Text is added to the output at specific positions. This dialog displays all the text settings:

- **Left**: Displays the distance between the left margin of the page and the text.
- **Bottom**: Displays the distance between the bottom margin of the page and the text .
- **Orientation**: Displays the orientation of the text.
- **Text**: Displays the actual text.

> **Note**
>
> The entered text might have been entered over multiple lines, so not all of the text will be displayed here. You might consider this a text entry preview of the text, rather than the complete text entry.

- **Condition**: Displays the condition which is used to determine if text element is to be included or not.
- **Add**: Click to open the Additional Text Settings dialog to add a new text entry.
- **Delete**: Click to delete the currently selected entry.
- **Edit**: Click to edit the currently selected entry using the Additional Text Settings dialog.
- **Duplicate**: Click to create a copy of the entry.

Additional Images

Images are added at specific positions, with optional dimension constrains. This dialog displays all the configured additional image settings:

- **Left**: Displays the distance between the left margin of the page and the image.
- **Bottom**: Displays the distance between the bottom margin of the page and the image .
- **Orientation**: Displays the orientation of the picture.
- **Filename**: Displays the selected image filename.
- **Condition**: Displays the condition which is used to determine if the image is to be included or not.
- **Add**: Click to open the "Additional Image Settings" on page 544 dialog to add a new text entry.

- **Delete**: Click to delete the currently selected entry.
- **Edit**: Click to edit the currently selected entry using the "Additional Image Settings" on page 544 dialog.
- **Duplicate**: Click to create a copy of the entry.

Additional Barcodes

Barcodes are added at specific positions. This dialog displays all the configured additional Barcode settings:

- **Left**: Displays the distance between the left margin of the page and the Barcode .
- **Bottom**: Displays the distance between the bottom margin of the page and the Barcode .
- **Orientation**: Displays the orientation of the Barcode .
- **Type**: Displays the type of Barcode that's added.
- **Text**: Displays a preview of the Barcode contents.
- **Condition**: Displays a preview of the condition.
- **Add**: Click to add a Barcode. Select from the list of Barcode types that appears. The Additional Barcode Settings page lists all the available Barcodes, and links to their options..
- **Delete**: Click to delete the currently selected Barcode entry.
- **Edit**: Click to edit the currently selected Barcode entry using the Additional Barcode Settings dialog.
- **Duplicate**: Click to create a copy of the barcode entry.

Additional OMR Marks

Optical Mark Recognition (OMR) marks are added at specific positions, with optional dimension constrains. This dialog displays all the configured additional image settings:

- **Left**: Displays the distance between the left margin of the page and the OMR mark.
- **Bottom**: Displays the distance between the bottom margin of the page and the OMR mark.
- **Orientation**: Displays the orientation of the OMR mark.
- **Condition**: Displays the condition which is used to determine if the OMR mark is to be included or not.
- **Add**: Click to open the "Additional OMR Mark Settings" on page 570 dialog to add a new text entry.

- **Delete**: Click to delete the currently selected entry.
- **Edit**: Click to edit the currently selected entry using the "Additional OMR Mark Settings" on page 570 dialog.
- **Duplicate**: Click to create a copy of the entry.

Additional Text Settings

The Additional Text Settings dialog displays the property of Text added in the "Additional Text" on page 540 page.

- **Position group**:
  - **Orientation**: Use the drop-down to select the orientation of the Text added to the page.
  - **Left**: Enter the distance between the left margin of the page and the Text, in either metric (cm/mm) inch (in) 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) 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.
  - **Font Size**: Enter the font size in points (pt).
  - **Bold**: Check to make the Text **bold**.
  - **Italic**: Check to make the Text *italic*.
  - **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 use the formatting options selected in the **Font group**.
  - **Add**: Click 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.
- **Condition**: Enter the condition which determines whether or not the Text will be added to the document at print time.
  For details on how to create a conditional, see the Conditionals page.

Additional Image Settings

The Additional Image dialog displays the properties of the image added in the "Additional Text" on page 540 page.

- **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)
  - **Left**: Enter the distance between the left margin of the page and the image, in either metric (cm/mm) inch (in) 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) or point (pt) values.
- **Filename**: Use the browse button to select an image.

> **Note**
>
> Transparent images are not supported.

- **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) or point (pt) values.
  - **Max Height**: Enter the absolute maximum height the image can be scaled to, in either metric (cm/mm) inch (in) 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.
  For details on how to create a conditional, see Conditionals page.

Barcode Options

When adding Barcodes added in the "Additional Text" on page 540 page you can select from a series of Barcodes. The options for these are described in the following pages:

- "Codabar Settings" on page 547
- "Code 128 Settings" on page 549
- "Code 39 Settings" on page 551
- "Additional Datamatrix Settings" on page 553
- "Additional EAN 128 Settings" on page 555
- "Additional EAN 13 Settings" on page 556
- "Additional EAN 8 Settings" on page 558
- "Additional Interleave 2 of 5 Settings" on page 560
- "Additional PDF417 Settings" on page 562
- "Additional QR Code Settings" on page 564
- "Additional UPC A Settings" on page 566
- "Additional UPC E Settings" on page 568

Standard Barcode Settings

- **Position group**:
  - **Orientation**: Use the drop-down to select the orientation of the Barcode added to the page.
  - **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.
- **Type group**:
  - **Type**: Use the drop-down to select which Barcode type to use.
  - **Height**: Enter the Barcode height in either metric (cm/mm) inch (in) or point (pt) values. Not available for certain Barcode types.
- **Text**: Enter the text used to generate the Barcode.

    - **Add**: Click to display a list of variable data that can be added to the Barcode. This includes metadata fields added in the 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 the Conditionals page.

Codabar Settings

Use the following options to configure the output barcode settings:

Position group:

- **Orientation**: Use the drop-down to select the orientation of the barcode added to the page.
- **Left**: Enter the distance between the left margin of the page and the barcode, in inches (in).
- **Bottom**: Enter the distance between the bottom margin of the page and the barcode, in inches (in).

Barcode-specific properties:

- **Height**:
- **Module Width**: Specifies the width of the narrow bars in centimeters. Changing this value to higher value will make the barcode bigger when Scale is set to None.
- **Bar width ratio**:
- **Default start symbol**: Use the drop-down to select the start character for the barcode, which defines the encoding mode.
- **Default stop symbol**: Use the drop-down to select the stop character for the barcode, which defines the encoding mode.

- **Print human readable text:** Check to add a textual version of the barcode data. Not available for 2d barcodes such as QR and PDF417.
  - **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.
  - **Font size**: Enter a font size for the human readable text.
- **Text**: Enter the text used to generate the barcode.
  - **Add**: Click to display a list of variable data that can be added to the barcode. This includes metadata fields added in the Metadata Options, as well as some

information fields.

- **Condition**: Enter the condition by which the barcode is shown on the page.
  - **Add**: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

Code 128 Settings

Use the following options to configure the output barcode settings:

Position group:

- **Orientation**: Use the drop-down to select the orientation of the barcode added to the page.
- **Left**: Enter the distance between the left margin of the page and the barcode, in inches (in).
- **Bottom**: Enter the distance between the bottom margin of the page and the barcode, in inches (in).

Barcode-specific properties:

- **Height**:
- **Module Width**: Specifies the width of the narrow bars in centimeters. Changing this value to higher value will make the barcode bigger when Scale is set to None.

- **Print human readable text:** Check to add a textual version of the barcode data. Not available for 2d barcodes such as QR and PDF417.
  - **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.
  - **Font size**: Enter a font size for the human readable text.
- **Text**: Enter the text used to generate the barcode.
  - **Add**: Click to display a list of variable data that can be added to the barcode. This includes metadata fields added in the Metadata Options, as well as some information fields.
- **Condition**: Enter the condition by which the barcode is shown on the page.

- **Add**: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

Code 39 Settings

Use the following options to configure the output barcode settings:

Position group:

- **Orientation**: Use the drop-down to select the orientation of the barcode added to the page.
- **Left**: Enter the distance between the left margin of the page and the barcode, in inches (in).
- **Bottom**: Enter the distance between the bottom margin of the page and the barcode, in inches (in).

Barcode-specific properties:

- **Height**:
- **Use extended character set**: Check to use the Code 39 Extended character set.
- **Module Width**: Specifies the width of the narrow bars in centimeters. Changing this value to higher value will make the barcode bigger when Scale is set to None.
- **Bar width ratio**:
- **Checksum**: Use the drop-down to select how to deal with the barcode checksum:
  - **Ignore**:
  - **Auto**:
  - **Check**:
  - **Add**:
- **Print human readable text:** Check to add a textual version of the barcode data. Not available for 2d barcodes such as QR and PDF417.
  - **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.
  - **Font size**: Enter a font size for the human readable text.
- **Text**: Enter the text used to generate the barcode.
  - **Add**: Click to display a list of variable data that can be added to the barcode. This includes metadata fields added in the Metadata Options, as well as some

information fields.

- **Condition**: Enter the condition by which the barcode is shown on the page.
  - **Add**: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

Additional Datamatrix Settings

Use the following options to configure the output barcode settings:

Position group:

- **Orientation**: Use the drop-down to select the orientation of the barcode added to the page.
- **Left**: Enter the distance between the left margin of the page and the barcode, in inches (in).
- **Bottom**: Enter the distance between the bottom margin of the page and the barcode, in inches (in).

Barcode-specific properties:

- **Module Width**: Specifies the width of the narrow bars in centimeters. Changing this value to higher value will make the barcode bigger when Scale is set to None.
- **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
  - **Text**: is used to encode data that mainly contains numbers and lowercase
  - **Base256**: used to encode 8 bit values
  - **None**: Does not use any encoding.
  - **Auto Detect**: Automatically detect the data content and encodes using the most appropriate method.
- **Format**: Use the drop-down to select the data matrix format (its size).

- **Print human readable text:** Check to add a textual version of the barcode data. Not available for 2d barcodes such as QR and PDF417.
  - **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.
  - **Font size**: Enter a font size for the human readable text.
- **Text**: Enter the text used to generate the barcode.

  - **Add**: Click to display a list of variable data that can be added to the barcode. This includes metadata fields added in the Metadata Options, as well as some information fields.
- **Condition**: Enter the condition by which the barcode is shown on the page.
  - **Add**: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

Additional EAN 128 Settings

Use the following options to configure the output barcode settings:

Position group:

- **Orientation**: Use the drop-down to select the orientation of the barcode added to the page.
- **Left**: Enter the distance between the left margin of the page and the barcode, in inches (in).
- **Bottom**: Enter the distance between the bottom margin of the page and the barcode, in inches (in).

[TBD]

- **Print human readable text:** Check to add a textual version of the barcode data. Not available for 2d barcodes such as QR and PDF417.
  - **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.
  - **Font size**: Enter a font size for the human readable text.
- **Text**: Enter the text used to generate the barcode.
  - **Add**: Click to display a list of variable data that can be added to the barcode. This includes metadata fields added in the Metadata Options, as well as some information fields.
- **Condition**: Enter the condition by which the barcode is shown on the page.
  - **Add**: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

Additional EAN 13 Settings

Use the following options to configure the output barcode settings:

Position group:

- **Orientation**: Use the drop-down to select the orientation of the barcode added to the page.
- **Left**: Enter the distance between the left margin of the page and the barcode, in inches (in).
- **Bottom**: Enter the distance between the bottom margin of the page and the barcode, in inches (in).

Barcode-specific properties:

- **Height**:
- **Module Width**: Specifies the width of the narrow bars in centimeters. Changing this value to higher value will make the barcode bigger when Scale is set to None.
- **Checksum**: Use the drop-down to select how to deal with the barcode checksum:
  - **Ignore**:
  - **Auto**:
  - **Check**:
  - **Add**:
- **Print human readable text:** Check to add a textual version of the barcode data. Not available for 2d barcodes such as QR and PDF417.
  - **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.
  - **Font size**: Enter a font size for the human readable text.
- **Text**: Enter the text used to generate the barcode.
  - **Add**: Click to display a list of variable data that can be added to the barcode. This includes metadata fields added in the Metadata Options, as well as some information fields.
- **Condition**: Enter the condition by which the barcode is shown on the page.

- **Add**: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

Additional EAN 8 Settings

Use the following options to configure the output barcode settings:

Position group:

- **Orientation**: Use the drop-down to select the orientation of the barcode added to the page.
- **Left**: Enter the distance between the left margin of the page and the barcode, in inches (in).
- **Bottom**: Enter the distance between the bottom margin of the page and the barcode, in inches (in).

Barcode-specific properties:

- **Height**:
- **Module Width**: Specifies the width of the narrow bars in centimeters. Changing this value to higher value will make the barcode bigger when Scale is set to None.
- **Bar width ratio**:
- **Checksum**: Use the drop-down to select how to deal with the barcode checksum:
  - **Ignore**:
  - **Auto**:
  - **Check**:
  - **Add**:
- **Print human readable text:** Check to add a textual version of the barcode data. Not available for 2d barcodes such as QR and PDF417.
  - **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.
  - **Font size**: Enter a font size for the human readable text.
- **Text**: Enter the text used to generate the barcode.
  - **Add**: Click to display a list of variable data that can be added to the barcode. This includes metadata fields added in the Metadata Options, as well as some information fields.

- **Condition**: Enter the condition by which the barcode is shown on the page.
    - **Add**: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

Additional Interleave 2 of 5 Settings

Use the following options to configure the output barcode settings:

Position group:

- **Orientation**: Use the drop-down to select the orientation of the barcode added to the page.
- **Left**: Enter the distance between the left margin of the page and the barcode, in inches (in).
- **Bottom**: Enter the distance between the bottom margin of the page and the barcode, in inches (in).

Barcode-specific properties:

- **Height**:
- **Module Width**: Specifies the width of the narrow bars in centimeters. Changing this value to higher value will make the barcode bigger when Scale is set to None.
- **Bar width ratio**:
- **Checksum**: Use the drop-down to select how to deal with the barcode checksum:
  - **Ignore**:
  - **Auto**:
  - **Check**:
  - **Add**:
- **Print human readable text:** Check to add a textual version of the barcode data. Not available for 2d barcodes such as QR and PDF417.
  - **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.
  - **Font size**: Enter a font size for the human readable text.
- **Text**: Enter the text used to generate the barcode.
  - **Add**: Click to display a list of variable data that can be added to the barcode. This includes metadata fields added in the Metadata Options, as well as some information fields.

- **Condition**: Enter the condition by which the barcode is shown on the page.
    - **Add**: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

Additional PDF417 Settings

Use the following options to configure the output barcode settings:

Position group:

- **Orientation**: Use the drop-down to select the orientation of the barcode added to the page.
- **Left**: Enter the distance between the left margin of the page and the barcode, in inches (in).
- **Bottom**: Enter the distance between the bottom margin of the page and the barcode, in inches (in).

Barcode-specific properties:

- **Module Width**: Specifies the width of the narrow bars in centimeters.
- **Row height**: Defines the height of the bars for a single row measured in pixels drawn
- **Width to height ratio**:
- **Mode**: Use the drop-down to set the compaction mode
  - **Numeric**: more efficient mode for encoding numeric data
  - **Text**: allows all printable ASCII characters to be end coded (values 32 to 126 and some additional control characters)
  - **Binary**: allows any byte value to be encoded
- **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) and level 8 (maximum error correction). Recommended error correction levels are between level 2 and 5, but the optimal value depends on 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 3 to 30.

- **Print human readable text:** Check to add a textual version of the barcode data. Not available for 2d barcodes such as QR and PDF417.
  - **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.
    - **Font size**: Enter a font size for the human readable text.
- **Text**: Enter the text used to generate the barcode.
    - **Add**: Click to display a list of variable data that can be added to the barcode. This includes metadata fields added in the Metadata Options, as well as some information fields.
- **Condition**: Enter the condition by which the barcode is shown on the page.
    - **Add**: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

Additional QR Code Settings

Use the following options to configure the output barcode settings:

Position group:

- **Orientation**: Use the drop-down to select the orientation of the barcode added to the page.
- **Left**: Enter the distance between the left margin of the page and the barcode, in inches (in).
- **Bottom**: Enter the distance between the bottom margin of the page and the barcode, in inches (in).

Barcode-specific properties:

- Size by:
  - By area:
  - By module width:
- Size:
- Minimum module width:
- **Encoding**: Define the encoding of the barcode:
  - **Auto**: when selected the library determines the encoding based on the supplied string.
  - **Numeric**: 7089 numerical characters.
  - **Alphanumeric**: 4296 alphanumerical characters.
  - **Byte**: 2953 characters.
  - **Kanji**: 1817 characters.
- **Version**: There are 40 sizes of QA codes, select the preferred version for the QR code.
- **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, which makes it larger.
- Use ECI for encoding messages as bytes:
- Multi-part QR Code:
  - Part:
  - of:

- **Use FNC1**: Check to enable FNC1.
  - **Position**: 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 ID**: Enter the QR-Code Format application indicator used by Industry FNC option.
- **Print human readable text:** Check to add a textual version of the barcode data. Not available for 2d barcodes such as QR and PDF417.
  - **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.
  - **Font size**: Enter a font size for the human readable text.
- **Text**: Enter the text used to generate the barcode.
  - **Add**: Click to display a list of variable data that can be added to the barcode. This includes metadata fields added in the Metadata Options, as well as some information fields.
- **Condition**: Enter the condition by which the barcode is shown on the page.
  - **Add**: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

Additional UPC A Settings

Use the following options to configure the output barcode settings:

Position group:

- **Orientation**: Use the drop-down to select the orientation of the barcode added to the page.
- **Left**: Enter the distance between the left margin of the page and the barcode, in inches (in).
- **Bottom**: Enter the distance between the bottom margin of the page and the barcode, in inches (in).

Barcode-specific properties:

- **Height**:
- **Module Width**: Specifies the width of the narrow bars in centimeters. Changing this value to higher value will make the barcode bigger when Scale is set to None.
- **Checksum**: Use the drop-down to select how to deal with the barcode checksum:
  - **Ignore**:
  - **Auto**:
  - **Check**:
  - **Add**:
- **Print human readable text:** Check to add a textual version of the barcode data. Not available for 2d barcodes such as QR and PDF417.
  - **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.
  - **Font size**: Enter a font size for the human readable text.
- **Text**: Enter the text used to generate the barcode.
  - **Add**: Click to display a list of variable data that can be added to the barcode. This includes metadata fields added in the Metadata Options, as well as some information fields.
- **Condition**: Enter the condition by which the barcode is shown on the page.

- **Add**: Click to display a list of metadata fields, information fields to add, or common expressions to the condition.

Additional UPC E Settings

Use the following options to configure the output barcode settings:

Position group:

- **Orientation**: Use the drop-down to select the orientation of the barcode added to the page.
- **Left**: Enter the distance between the left margin of the page and the barcode, in inches (in).
- **Bottom**: Enter the distance between the bottom margin of the page and the barcode, in inches (in).

Barcode-specific properties:

- **Height**:
- **Module Width**: Specifies the width of the narrow bars in centimeters. Changing this value to higher value will make the barcode bigger when Scale is set to None.
- **Checksum**: Use the drop-down to select how to deal with the barcode checksum:
  - **Ignore**:
  - **Auto**:
  - **Check**:
  - **Add**:
- **Print human readable text:** Check to add a textual version of the barcode data. Not available for 2d barcodes such as QR and PDF417.
  - **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.
  - **Font size**: Enter a font size for the human readable text.
- **Text**: Enter the text used to generate the barcode.
  - **Add**: Click to display a list of variable data that can be added to the barcode. This includes metadata fields added in the Metadata Options, as well as some information fields.
- **Condition**: Enter the condition by which the barcode is shown on the page.

- **Add**: 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 Text" on page 540 page.

These OMR marks differ from 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 support by a HCF, or they can be used for any non-inserter related post processing driven by OMR marks

- **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.
  - **Left**: Enter the distance between the left margin of the page and the OMR Mark, in either metric (cm/mm) inch (in) 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) 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 determined. 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 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 Conditionals 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 Conditionals 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.

PDF Options

The PDF Options page is shown only when a PDF Print output type is selected in the Print Options dialog.

- **PDF Options Group**
    - **PDF Type**: Use the drop-down to specify which format the PDF should be generated in. These options are standard PDF, archive format PDF (PDFA-1b), graphics format PDF (PDF-X4) and variable data printing format PDF (PDF-VT).
    - **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 ignored for PDFA and PDF-X4 output, as fonts are always embedded in those output types.

- **Add Digital Signature Group**: Check to enable the integration of a digital signature into the PDF.
  A digital signature identifies the person signing a document, similarly to a conventional handwritten signature. Unlike a handwritten signature, a digital signature is difficult to forge as it contains encrypted information which is unique to the signer and which can be password protected and verifiable.
  - **All Keystores**:
    Here you can choose from existing digital signatures, or select new ones.
    - **Name**: The user-defined name of the keystore.
    - **File**: The file path and name to the keystore file.

    This is where you select keystore values.

    - **New**: Click to open the Key Store dialog to add a new keystore to the list.
    - **Duplicate**: Click to make a copy of the currently selected keystore.
    - **Edit**: Click to edit the currently selected keystore in the Key Store dialog.
    - **Delete**: Click to delete the currently selected keystore.
    - **Move Up**: Click to move the currently selected keystore up.
    - **Move Down**: Click to move the currently selected keystore down.
  - **All Signatures**: Displays a list of signatures to add to the PDF output.
    - **Name**: The user-defined name of the signature.
    - **File**: The file path and name to the signature file.
    - **Alias**: The user-defined alias for the signature.
    - **New**: Click to open the PDF Signature dialog to add a new signature to the list.
    - **Duplicate**: Click to make a copy of the currently selected signature.
    - **Edit**: Click to edit the currently selected signature in the PDF Signature dialog.
    - **Delete**: Click to delete the currently selected signature.
    - **Move Up**: Click to move the currently selected signature up.
    - **Move Down**: Click to move the currently selected signature down.

Keystore

The security certificate Keystore dialog appears when adding or editing a keystore from the "PDF Options" on page 593 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.
- PKCS#12
- PKCS#11

> **Note**
>
> PKCS#11 requires an extra plug-in not included in the PReS Connect installation.

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.
- **Keystore properties group**:
  - **Type**: Use the drop-down to select the appropriate type of the keystore format the file is: JKS, PKCS11, PKCS12.
  - **Provider**: Enter the provider of the keystore.
    - "SUN" for JKS
    - "SunJSSE" for PKCS#12
    - "IAIK PKCS#11:1" for PKCS#11
  - **Password**: Type in the password that secures the keystore, if the keystore is password protected.
  - **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.
- **Properties file group**:
  - **File**: Load optional keystore properties file. Could be used to store the password in a file.

PDF Signature

The PDF Signature dialog appears when adding or editing a signature from the "PDF Options" on page 593 page.

- **Name**: Enter a name that describes the signature entry.
- **Keystore**: Use the drop-down to select which keystore the signature is pulled from. These keystores are set in the "Keystore" on page 595 dialog, called from the "PDF Options" on page 593 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.
    - **Handler**: The PDF reader plugin used to interpret the signature data. It should be left at its default setting (Adobe.PPKLite) unless time-stamping is desired, in which case "Adobe.PPKMS" is likely the best option.
- **Key group**: Refers to a key from the keystore.
    - **Alias**: The user-friendly name of the key
    - **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).
- **Apply Time Stamping Authentication group**: Check to enable time stamping authentication.

> **Note**
>
> Not available for signatures set to use Adobe.PPKLite Handler.

- **URL**: Select the Time Stamp Authority (TSA) URL address.
- **Account**: Account name specific to the TSA server chosen.
- **Password**: Password specific to the TSA server chosen.
- **Repeat Password**: Repeat of password.

- **Visible Signature group**: Check to add a visible signature to the PDF file.
  - **X**: Enter the horizontal distance between the left side of the page and the left side of the signature, in points (pt).
  - **Y**: Enter the vertical distance between the top of the page and the top of the signature, in points (pt).
  - **Width**: Enter the desired width of the signature, in points (pt).
  - **Height**: Enter the desired height of the signature, in points (pt).

Job Creation Presets

The Job Creation Setting dialog displays a list of available presets and a summary of their settings. Presets can also be edited from this dialog.

- **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 Data Filtering Options are set.
    - **Has Sorting**: Becomes Yes if any Sorting Options are set.
    - **Has Grouping**: Becomes Yes if grouping options are set in the Grouping and Splitting Options.
    - **Page Count Splitting**: Becomes Yes if page count splitting is used in the Grouping and Splitting Options.
    - **Slip Sheets**: Becomes Yes if a slip sheet is set in the Grouping and Splitting Options.
- **Options Group**: These options are checked, or not, depending on the selected preset chosen in the Configuration name.

    - **Use Grouping**: Check to activate the Grouping and Splitting Options page of the wizard.
    - **Apply filtering and sorting to record selection**: Check to activate the Data Filtering Options page of the wizard.
    - **Include Metadata (PDF and AFP only)**: Check to activate the Metadata Options page of the wizard.
    - **Override Finishing Options**: Check to activate the Finishing Options.
- **Next**: Click to go to the next page of the Job Creation Wizard, Data Filtering Options
- **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.

Finishing Options

Use this dialog to force specific finishing options, instead of using finishing options that were set in the Template's Media and Section options.
This is only applied when producing print output. It does not modify the original finishing options in either the Section or the Template.

- **Ignore section level finishing**: Check to override finishing options at the document level only.
- **Section to edit**: Use the drop-down to select which Section to apply the options below. The Document level is also listed to edit document-level finishing.
    - **Settings:** Click the settings button to bring up control options:
        - **Reload:** Restores the current section's properties to the default values set in the template for this section
        - **Reload All:** Restores all section properties to the default values set in the template for each section.
        - **Apply finishing from:** Displays a list of available sections. Clicking on a section name loads that section's properties into the current section to edit.
        - **Apply current finishing to all sections:** Applies the current properties to all sections.
- **Binding** group:
    - **Style**: What type of Binding to request on the printer. This includes Stapled, Glued, Stitched, Ring, and various other options..
    - **Side**: Sets the side of the paper that the Binding is to occur.
    - **Location**: Sets where the binding is to occur, if applicable.
      The selections available here are 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 actually available to you 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.
    - **Style** The style of hole punches for the selected Binding option.
    - **Pattern Catalog ID**: The Catalog ID of the selected Binding option.

Data Filtering Options

The data filtering page is used to filter records and prevent them from being printed. Conditions are evaluated on each record.

Data Selection Filter

- **Grouping**: Displays the type of line, either a Rule or a rule Grouping. The root of each group of rule is a drop-down selector that defines how the rules inside the grouping work together, which is either to make *any of the rules*or*all of the rules*have to be true for the group to be true.
- **Field**: Use the drop-down to select the field on which to make the comparison.
- **Operator**: Use the drop-down to select the comparison operator for the condition.
- **Value**: Type in a value for the comparison.
- **Add**: Click to add a new line to list. Different options are available in this menu, such as filtering by field, media and finishing properties, or document length.

- **Add a new nested rule group**: Click to add a new grouping at the current level.
- **Delete**: Click to delete the currently selected rule or group. **Note**: deleting a group deletes all rules under it, and this action cannot be undone.
- **Group selected rules as nested rules**: Click to create a group with the currently selected rules.
- **Merge selected rules/ruleset to parent rules**: Click to move the currently selected rule (s) to the parent group.

Preview

This box displays a textual representation of the conditions set in the data filtering.

Sorting Options

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**: Click to move the currently selected row up in the list.
  - **Move down**: Click to move the currently selected row down in the list.
- **Use external sort**: Sort the records using an external sorting software. A CSV file is exported, sorted by the external application and the sorted CSV file is returned and integrated, with the records now sorted according to the new order in the CSV file.
  - **Command**: Enter the full path to the executable that will sort the CSV file.
  - **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: Windows (CRLF), Linux (LF) or Mac (CR).

    - **Character Set**: Use the drop-down to select which character set to use when encoding the CSV file.
    - **Fields to export**: Lists the fields to export in the CSV file. Click the left arrow button to add a field or the right arrow to remove the field from the list.
    - **Available Fields**: Lists the fields available in the selected data mapping configuration that can be used to sort the records.

Grouping Options

The Grouping options separates the job output into multiple blocks that can then be physically separated using split sheets in the printer.

- **Grouping Tabs**: Jobs can be grouped at three different levels, each of which is each contained in a tab in this section. The groups/tabs are:
    - **Job** Grouping Fields
    - **Job Segment** Grouping Fields
    - **Document Set** Grouping Fields

  All the Fields available to be used for Grouping are contained within the *Available Fields* box in each tab. Fields that you want to use for Grouping need to be added to the *Selected Fields* box via the arrows between the two boxes. Simply select the Field(s) you want to move and click the 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.

- **Page Break Grouping**: Check to enable page break grouping, which separates different groups by the number of pages they contain. For example, enabling the Document Set Grouping Level and creating a page range from 1-5 and 6 to Largest, will create two groups. The first will contain all document sets of 1 to 5 pages, the second will contain any document set of 6 or more pages.
    - **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.
    - **Grouping list**: 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".
  - **Generate page break ranges in reverse order**: Reverses the order of the groups created. By default, grouping will be from smallest to largest. Checking this option creates groups from largest to smallest.
  - **Generate page break range groups after normal grouping**: Check this option to first group using the levels above, following which page break grouping are applied. This creates two different levels of grouping, applied in order.

Metadata Options

The Metadata Option page defines metadata tags that will be added to the output file when producing PDF and AFP output in the Output Creation Presets. Metadata tags are ignored in all other output types. The tags are added to each of the levels, as indicated by the tabs on top: **Job**, **Job Segment**, **Document**, **Document Set**, and **Page** Tags.

In each of these levels, a list of tags is available:

- **Always create meta data for this level even when fields are selected**: Select 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**: Name of the metadata 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**: Displays the type of field being used - either Text or Data Field.
- **Source**: For Data Fields only. The Field name from the data mapping configuration whose value will be used for this tag.
- ➕ **Add Field**: Click to add a new tag to the current level. The Field Selection dialog appears. Select either **Add field meta data** or **Add text meta data**.
  When adding field meta data select a field name from the Field List and click OK to add it as a tag of the same name.
- ➖ **Delete Field**: Click to delete the currently selected tag.
- ⬆ **Move Up**: Click to move the currently selected tag one position up.
- ⬇ **Move Down**: Click to move the currently selected tag one position down.

Output Creation Settings

The Output Creation Settings dialog displays a list of available presets and a summary of their settings. Presets can also be edited from this dialog.

- **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 Output Presets using a Browse dialog.
- **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
  - Inserter: Indicates whether Inserter Marks have been added in the Inserter Marks dialog. Expand to see which HCF (Hardware Configuration File) is loaded.
  - Imposition: Indicates if Imposition has been set in the "Imposition Options" on page 589 dialog. Expand to see the specific imposition settings.
  - Has custom printer settings: Indicates if custom printer settings have been set in the Printer Settings 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.

Click Next in this dialog to see the Print Options window where output creation settings are selected.

Print Options

This section describes the Print Options, which is the first page of the **Advanced Print Wizard** as well as the Output Creation Settings Preset .

This page is the most important of the Advanced Print Wizard.
The pages shown throughout the Wizard are determined by the selections made on this page.

The choices can be broken down as follows:

- **Printer** section:
  - **Model**: Use the drop-down to select the printer language / output type that will be generated.
    Connect output options cover a range of industry standard print output types.
    These include AFP, IPDS, PCL, PDF and PostScript (including PPML, VIPP and

VPS variants), with a range of quality settings available.

> **Note**
>
> By default, Connect displays only the PDF output option, but other print output types can be added to the Printer Model drop down list via the Gear button

For more information on how to do this, see "Adding print output types to the Print Wizard" on page 308.

- **Output Options** section:
  - **Output Local** checkbox: 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 filename.
    - **Directory**: Select to output to a local folder on the machine.
      - **Job Output Mask**: The name of the file that will output.
        You can use ${template} as a variable for the name of the Designer Template used to generate the output.
      - **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.

    - **Windows Printer**: Select to send the Print Job to a Printer Queue. The job is rendered as a PDF before being printed through the Windows driver.
        - **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 (PDF output only)**:
        - **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**:
            - **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 resized.
            - **Shrink to printable area**: Select to shrink any page to fit the page area. Pages smaller than the paper size are not resized.
- **Production Options:**
    - **Booklet Imposition** checkbox: Check to tell the printer to generate a booklet for the print output. Booklet options are set in the "Booklet Options" on page 588 page. This option is unselected by default unless selected in the Designer "Print Section Properties" on page 469.
    - **Cut and Stack Imposition** checkbox: Check to enable Cut & Stack Imposition, which is set in the "Imposition Options" on page 589 page.
    - **Add Inserter marks** checkbox: Check to enable inserter mark functionality, which is set in the "Inserter Options" on page 591 page.
    - **Override Finishing options** checkbox: Check to configure custom "Finishing Options" on page 578, such as binding.
    - **Print virtual stationery** checkbox: Check to enable virtual stationery in the output.
    - **Use grouping** checkbox: Check to configure grouping of output into jobs, job segments or document sets. See "Grouping Options" on page 581.

- **Include meta data** checkbox: Check to add meta data to the output. This can be done at Job, Job Segment, Document, Document Set and Page level. See "Metadata Options " on page 582.
    - **Separation**: Check to activate the "Separation Options" on the next page page of the wizard.
    - **Add additional content** checkbox: Check to activate the "Additional Text" on page 540 page of the wizard.
- **Records** section:
    - **Record Range**: Allows selection of a range of records or a custom selection.
      You can specific 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.
- **Copies** section:
    - **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** section:
  This section is valid for AFP, IPDS and PCL only. It applies to elements within the record that are shades of gray, rather than black or white.
    - **Black Threshold Percentage**: The percentage of shading at which the element will appear as full black, rather than dark gray.
    - **White Threshold Percentage**: The percentage at which the element will appear as full white, rather than light gray.

Printer Settings

The Printer Settings page defines options on the printer. It is available for PostScript (and the VIPP and VPS variants of PostScript) only.

- **Use Media Position** checkbox: Enables the *Position* column and removes the *Weight*, *Type* and *Color* columns, to simplify the selection of media, using only the Media Position option.

- Tray selection columns:

- **Media**: Lists the medias 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.
- **Weight**: Enter a weight for the paper.
- **Type**: Use the drop-down to select which type of stock to use on the printer.
- **Color**: Use the drop-down to select which color the paper should be on the printer.

Separation Options

This page defines how to separate the jobs using subsets, slip sheets, or jogging.

- **Sheet Count Splitting** group.
  This group allows for the splitting of output based upon a pre-determined number of pages
  - **Split**: Use the drop-down to select how to split.
    - **None**: Select to ignore sheet count splitting entirely.
    - **At exactly**: Select to create a split at a specific sheet number.
  - **Every**: Enter the number of sheets at which to split the output.
- **Separation Settings** group.
  This setting is only available if no Sheet Count Split were 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.

Booklet Options

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** sections 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.

Imposition Options

The imposition options page sets the repetition, order, margins and marks for imposition in the output.

- **Sheet Size** group:
    - **Final Media Size**: Use the drop-down to select the size of the media where the output is printed. The size of the media should be equivalent to the initial section size multiplied by the number of repetitions, added with the margins and spaces between the repetitions.
      If *Custom* media size is selected, enter the custom **Width** and **Height** values.

      > **Note**
      >
      > The Sheet Size cannot be altered if a Cover Page was selected in the "Booklet Options" on the previous page Page.

    - **Sheet Rotations**: Select aspect ratio of media (Landscape or Portrait), or allow Connect to automatically determine the proper aspect ratio.
    - **Scale to fit**: Modify the size of each repetition to fit within the final media size.
    - **Rotate final output Sheet 180 degrees (upside down)**: Select to flip the output upside down.
- **Repetition** group:
  Allows selection of how many sections are to be placed, both Horizontally and Vertically. This is the total number of items, not the number of additional items being placed.

  > **Note**
  >
  > If *Booklet Binding* were selected, some of these settings will be determined by the options made within the "Booklet Options" on the previous page Page and they

cannot be altered here.

- **Space Between** group:
  Allows selection of the amount of blank space to add between each repetition.
- **Order** group:

> **Note**
>
> If *Booklet Binding* were selected, some of these settings will be determined by the options made within the "Booklet Options" on page 588 Page and they cannot be altered here.

  - **Page Order**: Select in which direction to go when adding sections to the output:
    - **Left to right, then top to bottom**
    - **Right to left, then top to bottom**
    - **Top to bottom, then left to right**
    - **Top to bottom, then right to left**
  - **Stack Depth**: Enter a stack depth or use the arrows to increment or decrement.
  - **Reverse Pages**: Select this option to reverse the order of pages.
    This would print the final record on the first page and the first record on the last page.
  - **Force simplex**: Select this option to make the output Simplex, rather than the imposition default of Duplex.
- **Bleed Margins** group:
  - **Top, Bottom, Left, Right**: Enter the bleed margins for each side of the page.
- **CropMarks** group:
  - **Type**: Use the drop-down to select the type of crop marks to add to the page.
  - **Offset**: How much separation (if any) to leave between the vertical and horizontal corner markings.
  - **Width**: Select the width of the crop mark lines.
  - **Length**: Select the Length of the crop mark lines.

Inserter Options

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 page influence the position of the markings set on the next page: **"Mark Position Options" on the facing page**.

- **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 the facing page 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.
    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".
- **Clear Background Area**: Check 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 a measure for the margins on each side of the OMR Marks.
- **Custom OMR mark sizing**: If supported by the chosen Mark Configuration you can select a Custom OMR size.
  You can 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.

Mark Position Options

This 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 the previous page 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 sections 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%

- 75% **Zoom Level**: Use the drop-down to select a predefined level or enter a zooming percentage.

PDF Options

The PDF Options page is shown only when a PDF Print output type is selected in the Print Options dialog.

- **PDF Options Group**
  - **PDF Type**: Use the drop-down to specify which format the PDF should be generated in. These options are standard PDF, archive format PDF (PDFA-1b), graphics format PDF (PDF-X4) and variable data printing format PDF (PDF-VT).
  - **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 ignored for PDFA and PDF-X4 output, as fonts are always embedded in those output types.

- **Add Digital Signature Group**: Check to enable the integration of a digital signature into the PDF.
  A digital signature identifies the person signing a document, similarly to a conventional handwritten signature. Unlike a handwritten signature, a digital signature is difficult to forge as it contains encrypted information which is unique to the signer and which can be password protected and verifiable.
  - **All Keystores**:
    Here you can choose from existing digital signatures, or select new ones.
    - **Name**: The user-defined name of the keystore.
    - **File**: The file path and name to the keystore file.

    This is where you select keystore values.

    - **New**: Click to open the Key Store dialog to add a new keystore to the list.
    - **Duplicate**: Click to make a copy of the currently selected keystore.
    - **Edit**: Click to edit the currently selected keystore in the Key Store dialog.

  - **Delete**: Click to delete the currently selected keystore.
  - **Move Up**: Click to move the currently selected keystore up.
  - **Move Down**: Click to move the currently selected keystore down.

- **All Signatures**: Displays a list of signatures to add to the PDF output.
  - **Name**: The user-defined name of the signature.
  - **File**: The file path and name to the signature file.
  - **Alias**: The user-defined alias for the signature.
  - **New**: Click to open the PDF Signature dialog to add a new signature to the list.
  - **Duplicate**: Click to make a copy of the currently selected signature.
  - **Edit**: Click to edit the currently selected signature in the PDF Signature dialog.
  - **Delete**: Click to delete the currently selected signature.
  - **Move Up**: Click to move the currently selected signature up.
  - **Move Down**: Click to move the currently selected signature down.

PDF Digital Signature Options

PDF Signature

The PDF Signature dialog appears when adding or editing a signature from the "PDF Options" on the previous page page.

- **Name**: Enter a name that describes the signature entry.
- **Keystore**: Use the drop-down to select which keystore the signature is pulled from. These keystores are set in the "Keystore" on the next page dialog, called from the "PDF Options" on the previous page 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.
  - **Handler**: The PDF reader plugin used to interpret the signature data. It should be left at its default setting (Adobe.PPKLite) unless time-stamping is desired, in which case "Adobe.PPKMS" is likely the best option.
- **Key group**: Refers to a key from the keystore.

- **Alias**: The user-friendly name of the key
- **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).

- **Apply Time Stamping Authentication group**: Check to enable time stamping authentication.

> **Note**
>
> Not available for signatures set to use Adobe.PPKLite Handler.

  - **URL**: Select the Time Stamp Authority (TSA) URL address.
  - **Account**: Account name specific to the TSA server chosen.
  - **Password**: Password specific to the TSA server chosen.
  - **Repeat Password**: Repeat of password.

- **Visible Signature group**: Check to add a visible signature to the PDF file.
  - **X**: Enter the horizontal distance between the left side of the page and the left side of the signature, in points (pt).
  - **Y**: Enter the vertical distance between the top of the page and the top of the signature, in points (pt).
  - **Width**: Enter the desired width of the signature, in points (pt).
  - **Height**: Enter the desired height of the signature, in points (pt).

Keystore

The security certificate Keystore dialog appears when adding or editing a keystore from the "PDF Options" on page 593 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.
- PKCS#12

- PKCS#11

> **Note**
>
> PKCS#11 requires an extra plug-in not included in the PReS Connect installation.

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.
- **Keystore properties group**:
  - **Type**: Use the drop-down to select the appropriate type of the keystore format the file is: JKS, PKCS11, PKCS12.
  - **Provider**: Enter the provider of the keystore.
    - "SUN" for JKS
    - "SunJSSE" for PKCS#12
    - "IAIK PKCS#11:1" for PKCS#11
  - **Password**: Type in the password that secures the keystore, if the keystore is password protected.
  - **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.
- **Properties file group**:
  - **File**: Load optional keystore properties file. Could be used to store the password in a file.

Print Manager

The **Print Manager** must be used when generating output for IPDS printers, but can also be used in conjunction with PCL and PostScript outputs.

IPDS printing requires a *print stream* and PReS Connect only produce *files*, thus this intermediary **Print Manager** is necessary to communicate to IPDS printers. It feeds the IPDS output to the printer, as a print stream, rather than as a file.

Neither PCL nor PostScript printers require a *print stream*, so such output files can be sent directly to a printer, from within Connect. You could use the **Print Manager** application, however, to extra print management functionality, such as organizing job priorities and job queues.

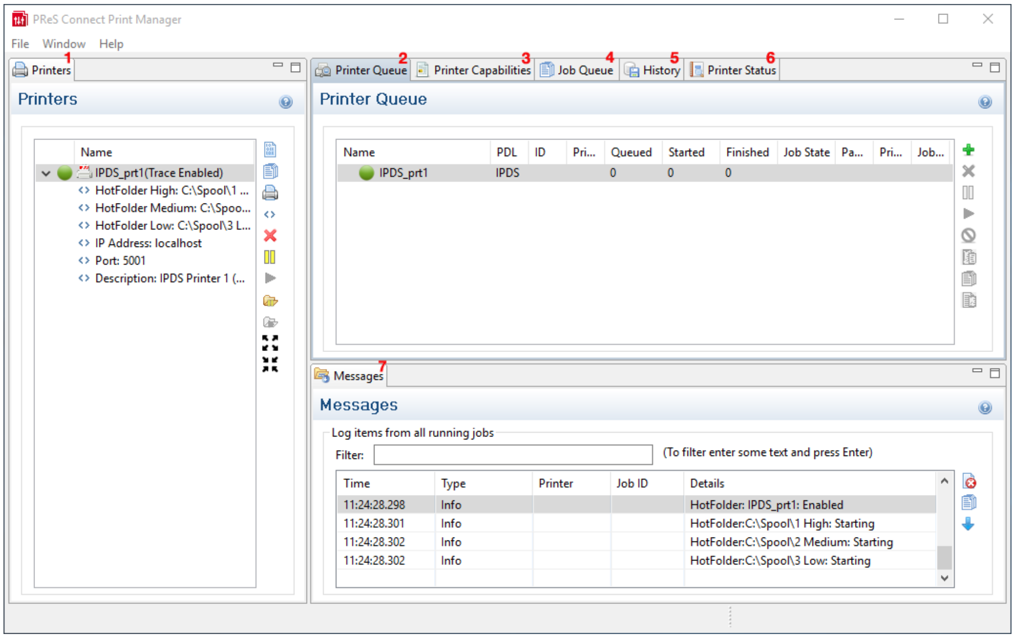
Print Manager Introduction Video

Print Manager usage

For a full description of how to use **Print Manager**, see "Print Manager Interface" below

Print Manager Interface

The Connect Print Manager Interface is broken up into multiple tabs. These can be re-organised and moved as desired, but appear by default as in the following image.



Printers

Name

- IPDS_prt1(Trace Enabled)
  - HotFolder High: C:\Spool\1 ...
  - HotFolder Medium: C:\Spoo...
  - HotFolder Low: C:\Spool\3 L...
  - IP Address: localhost
  - Port: 5001
  - Description: IPDS Printer 1 (...

Printer Queue

Name	PDL	ID	Pri...	Queued	Started	Finished	Job State	Pa...	Pri...	Job...
IPDS_prt1	IPDS			0	0	0				

Messages

Log items from all running jobs

Filter: (To filter enter some text and press Enter)

Time	Type	Printer	Job ID	Details
11:24:28.298	Info			HotFolder: IPDS_prt1: Enabled
11:24:28.301	Info			HotFolder:C:\Spool\1 High: Starting
11:24:28.302	Info			HotFolder:C:\Spool\2 Medium: Starting
11:24:28.302	Info			HotFolder:C:\Spool\3 Low: Starting

1. "Printers pane" below
2. "Printer Queue pane" on page 601
3. "Printer Capabilities pane" on page 602
4. "Job Queue pane" on page 603
5. "History pane" on page 603
6. "Printer Status pane" on page 604
7. "Messages pane" on page 605

Menu options

- **Windows > Preferences ...**: The PReS Connect Print Manager general settings can be changed via the **Windows > Preferences ...** menu option. Information on the preferences available can be found on this page: "Print Manager Preferences" on page 605.
- **Windows > Reset Perspective**: The PReS Connect Print Manager perspective can be reset via the **Windows > Reset Perspective** menu option. This returns all the Print Manager panes (which can be moved around as desired) to their original default positions.

Printers pane

The Printers pane displays a list of printers that have been added to the Print Manager.

The following information (all editable using the "Printer Settings Dialog" on the facing page) 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 two 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.
- **Hot Folders**: Displays the paths of any hot folders.
- **IP Address**: Displays the IP Address or machine name of the IPDS printer.
- **Port (IPDS Only)**: Displays the port of the printer communicates on.
- **LPR (PCL and PostScript Only)**: Displays the printing file type.
- **Description**: Displays printer description.

To the right of the printer list, the following controls are available:

- **Add job to selected printer**: Click to load an file directly and add it to the printer queue.
- **Add job with Page Range**: Click to load an file directly and add it to the printer queue while specifying the page range to print.
- **Add Printer**: Click to add a new printer using the "Printer Settings Dialog" on the facing page.
- **Edit Printer**: Click to edit the currently selected printer using the "Printer Settings Dialog" on the facing page.
- **Delete**: Click to delete the currently selected printer.
- **Pause**: Click to pause the currently selected printer's queue, which prevents any new jobs from being printed.
- **Resume**: Click to resume the currently selected printer's queue. Any job within the queue will start being printed.
- **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.
- **Resume HotFolder**: Click to resume capturing new jobs in the hot folder for the currently selected printer.
- **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.

Printer Settings Dialog

This dialog allows you to add or modify custom settings for printers. The settings available will differ between printer types. IPDS output has more options than the others.
Each printer setting section appears in it's own area, which can be folded up or down by clicking on the blue title bar atop each section.

The available entries include the following options:

- **Printer Details**
  - **Enable Printer**: Check to enable the printer settings.
    - **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**
      >
      > This will almost always be RAW.

    - **Hot Folders**: 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. Then enter the folders from which the Print Manager will capture print files automatically and send them to the appropriate Priority queue on the printer. There are **High**, **Medium** and **Low** priority queues available and each can be set here.
    - **Extended Logging Options (IPDS only)**: Defines logging of printer activities..
      - **Ack Pages:** The number of pages to send to the printer for printing before sending an acknowledgment request.
        Acknowledgment requests slow the printing process down, as the printer needs to process the request and return the information.
        The default Ack Pages number is 100 pages which means the Print

        Manager will send an acknowledgment request to the printer in 100 page increments until the end of the job. Given the overhead in processing such requests, the smaller the Page number, the slower the printing becomes.
        - **Create Trace File:** Select this to log all communications to a log file.
    - **End of Job Preferences (IPDS only):** Select these to distinguish job end.
        - **Edge Marks:** Select this to add an Edge Mark to the final page of a job. Used to indicate job separation in Continuous Form printing.
        - **Stack Receive Pages:** Add blank pages to the end of the job in order to push the completed job to the stacker.
    - **Error Handling Preferences (IPDS only):** Select these to **STOP** printing upon encountering the following errors:
        - **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:** Stops on all other optional errors.
- **Description**: Enter a description for the printer, if desired.

Printer Queue pane

The Printer Queue pane, similarly to the Job Queue pane, displays jobs. However in this case, it displays only jobs that are associated with a specific printer.

- Printer List
    - **Name**: Displays the name of the printer where the job is located.
    - **PDL**: Displays the type of printer language.
    - **ID**: The generated ID for the job. The first 2 letters of the ID indicate the origin of that job:
        - **HF**: Hot Folder capture.
        - **DR**: Dragged and dropped directly into the Print Manager.
        - **BR**: Added via a Browse button.
        - **RS**: Resubmitted to the printer.
    - **Priority**: Displays the priority in which the job is being printed: Low, Medium, High.

    - **Queued**: Displays the time that the job entered the printer queue, or the number of jobs queue on the printer.
    - **Started**: Displays the time that the job started being printed, or the number of started jobs on the printer.
    - **Finished**: Displays the time that the job has finished printing, or the number of finished jobs on the printer.
    - **Job State**: Displays the current job status such as Queued, Finished, Started, etc.
    - **Page Range**: Displays the range of pages to be printed.
    - **Printed Pages**: Displays the current number of printed pages in the current job, or the total number of printed pages on the printer.
    - **Job Pages**: Displays the total number of pages in the job.
- **Add Job**: Click to add a new job to the selected printer in the list.
- **Delete Job**: Click to delete the currently selected job.
- **Pause Job**: Click to temporarily pause the current job on the printer queue.
- **Start Job**: Click to start a job that has been paused.
- **Cancel Job**: Click to cancel a job.
- **Resubmit Job**: Click to restart a job from the beginning.
  Resubmit Range of Pages: Click to resubmit a selection of pages.
  Re-assign to Job Queue: Takes the job out of the Printer Queue and places it back in the generic Job Queue.

Printer Capabilities pane

The Printer Capabilities pane displays the technical capabilities of any printer that has been added to the "Printers pane" on page 598.

- **Filter**: Type in a filter for a specific capability name and press Enter to filter the results below.
- **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 on the right.

> **Note**
>
> Requesting a printer's capabilities can take some time, and not all printers will support it.

    - **Capabilities**: Each printer capability is listed under the Printer, in separate sections. Capabilities are read directly from the printer.

Job Queue pane

The Job Queue pane is a centralized location where all the jobs that are captured, added or sent to the Print Manager are listed, whether they have been printed or not.

- Filter: Type in a search word and hit Enter to filter the job queue. This searches the job name. Clear and press Enter to remove the filtering.
- **Job List**:
    - **Name**: The name of the job filename, as captured or added to the Print Manager.
    - **ID**: The generated ID for the job. The first 2 letters of the ID indicate the origin of that job:
        - **HF**: Hot Folder capture.
        - **DR**: Dragged and dropped directly into the Print Manager.
        - **BR**: Added via a Browse button.
        - **RS**: 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.

History pane

The History pane displays a history of all jobs that have been printed through the Print Manager.

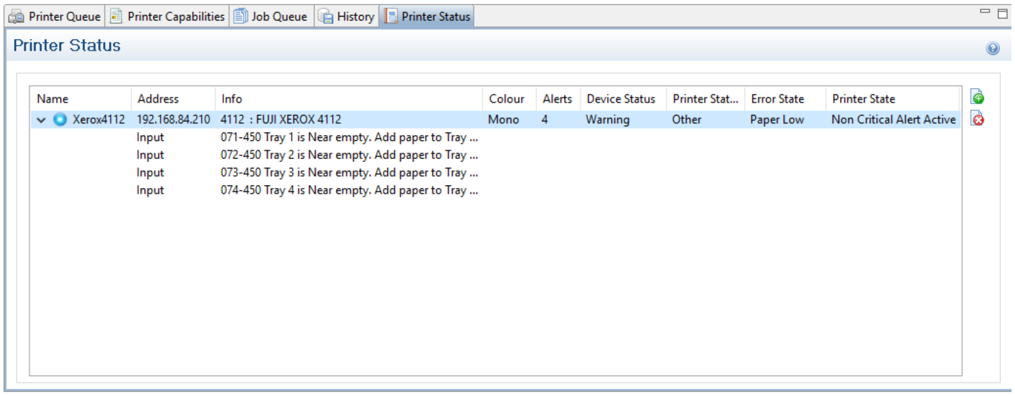
- **Filter**: Type in a search word and hit Enter to filter the job history. Clear and press Enter to remove the filtering.
- **History Log**:
    - **Date**: Displays the date and time when the job was started.
    - **Printer**: Displays the name of the printer the job was printed to.
    - **ID**: Displays the internal ID of the job.
    - **Job**: Displays the name of the job based on its filename.
    - **Path**: Displays the path from where the job was captured or added.
    - **Pages**: Displays the pages printed and the total number of pages in the job.

- **Status**: Displays the final status of the job, such as Cancelled or Printed.
- **Priority**: Displays in which queue priority the job was printed on the printer.

- **Copy selected items to clipboard**: Click to copy all messages to the clipboard to paste them in an external software.
- **Always select last log entry**: Click to toggle auto-scrolling so that the last message is always displayed when it appears.
- **Remove Job**: Click to delete job from history.

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/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:



Printer Status

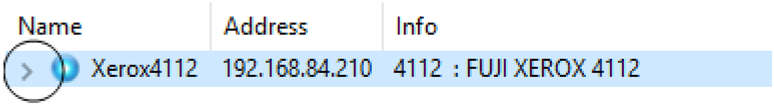
Name	Address	Info	Colour	Alerts	Device Status	Printer Stat...	Error State	Printer State
Xerox4112	192.168.84.210	4112 : FUJI XEROX 4112	Mono	4	Warning	Other	Paper Low	Non Critical Alert Active
	Input	071-450 Tray 1 is Near empty. Add paper to Tray ...						
	Input	072-450 Tray 2 is Near empty. Add paper to Tray ...						
	Input	073-450 Tray 3 is Near empty. Add paper to Tray ...						
	Input	074-450 Tray 4 is Near empty. Add paper to Tray ...						

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:

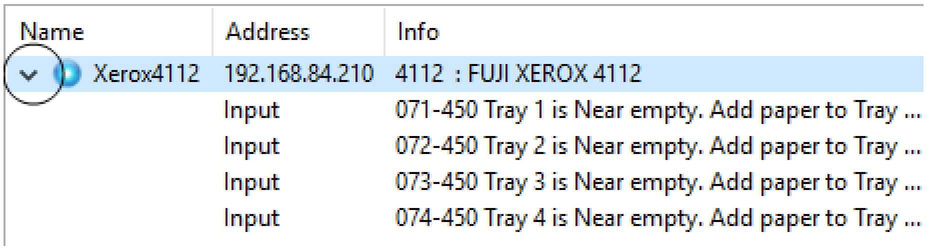
Once a printer is being monitored, the details can expanded and the individual printer trays displayed. These can be done as follows:

- **To expand the printer details**, press the right arrow besides the printer Name, as seen in the following image.



Name	Address	Info
Xerox4112	192.168.84.210	4112 : FUJI XEROX 4112

- **To reduce the printer details**, press the down arrow besides the printer Name, as seen in the following image.



Name	Address	Info
Xerox4112	192.168.84.210	4112 : FUJI XEROX 4112
	Input	071-450 Tray 1 is Near empty. Add paper to Tray ...
	Input	072-450 Tray 2 is Near empty. Add paper to Tray ...
	Input	073-450 Tray 3 is Near empty. Add paper to Tray ...
	Input	074-450 Tray 4 is Near empty. Add paper to Tray ...

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, that errors have occurred, etc.

- **Filter**: Type in a search word and hit Enter to filter the messages. Clear and press Enter to remove the filtering.
- **Message List**:
  - **Time**: Displays the timestamp for the message.
  - **Type**: Displays the type of message.
  - **Printer**: Displays the name of the relevant printer, if applicable.
  - **Job ID**: Displays the relevant Job ID, if applicable.
  - **Details**: Displays the message itself.
- **Clear Log Items**: Click to clear the current log. This does not delete log files.
- **Copy selected items to clipboard**: Click to copy all messages to the clipboard to paste them in an external software.
- **Always select last log entry**: Click to toggle auto-scrolling so that the last message is always displayed when it appears.

Print Manager Preferences

The Print Manager **Windows > Preferences ...** menu option allows the adjustment of Print Manager preferences.

There are three groups of preferences which can be altered, or set back to default settings at any time. These groups are:

1. "Language Choice" below
2. "General Settings" below
3. "SNMP Connection Settings" on the next page

Language Choice

Select the display language.
As of PReS Connect version 1.4.2 the languages supported are:

- German (DE)
- English (EN)
- Spanish (ES)
- French (FR)
- Italian (IT)
- Japanese (JA)
- Chinese Simplified (ZH)
- Chinese Traditional (ZH_HK, ZH_MO or ZH_TW)

General Settings

- **Hot Folder**: Defines a global hot folder that captures jobs and places them in the Job Queue pane. This does not automatically print the jobs, however.
    - **Enable Hot Folder**: Check to enable the global hot folder.
    - **Monitored Folder**: Enter a path for the hot folder, or click Browse to browse for it.
- **Session Retries**: The number of times Print Manager will try to reconnect to the printer before it stops trying and produces a timeout error in the messages.
- **Printer Timeout (sec)**: The startup time of the printer before the timeout is reported in the messages.
- **Printer Ack Timeout (sec)**: This is the time that Print Manager waits for a response from the printer after sending a response request.
- **Retry Delay**: This is the time in second that the Print Manager waits before retrying a job after the timeout.
- **Log Level**: Use the drop-down to select which log level to save. Higher log levels such as "Debug" give more helpful messages and details in the log but can impact processing time and resources.

- **Printer Preferences**:
  - **Hide disabled printers**: Check box to have any printers that have been set to disabled in the "Printer Settings Dialog" on page 600 not appear in the "Printers pane" on page 598.

SNMP Connection Settings

Options for printers that support Simple Network Management Protocol (SNMP)connection.

These include:

- **Connection attempts**: Defines the number of times to try a connection before abandoning the connection attempt,
- **Polling interval (secs)**: Defines the polling interval in seconds.
  This determines how often the Print Manager communicates with the printer. This setting applies to all communications with the printer. It impacts upon the **Connect attempts** (determining the interval between attempts) as well as how often the printer will be updated in the "Printer Status pane" on page 604.

Copyright Information

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.

Legal Notices and Acknowledgements

The license agreements for the associated open source third party components can be downloaded here.

This application uses the following third party components:

- **Apache Javascript Support (Rhino) version 1.7.3** which is licensed under the terms of the Mozilla License Version 1.1. The source code can be obtained from the following location: https://developer.mozilla.org/en-US/docs/RhinoDownload?redirect=no
- **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.
- **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 Version 2.0. Information on obtaining Gecko can be found on the following page: https://wiki.mozilla.org/Gecko:Getting_Started
- **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
- **Hamcrest Matchers** Copyright © 2000-2006, www.hamcrest.org. All rights reserved.
- **HyperSQL**, Copyright © 2001-2010, The HSQL Development Group. All rights reserved.
- **ICU4J 4.4.2** Copyright © 1995-2013 International Business Machines Corporation and others. All rights reserved.
- **Jacob Java Com Bridge** which is licensed under the terms of the GNU Lesser General Public License Version 2. The source code for this can be obtained from the following location: http://sourceforge.net/projects/jacob-project/files/jacob-project/

- **JavaCraft JSch** Copyright © 2002 - 2012 Atsuhiko Yamanaka, JCraft Inc. All rights reserved.
- **JavaSysMon** Copyright © 2009 ThoughtWorks, Inc. All rights reserved.
- **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
- **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
- **JGoodies Forms, JGoodies Binding and JGoodies Looks**, Copyright © 2002-2013 JGoodies Software GmbH. All rights reserved.
- **JNA Version 3.5.1** which is distributed under the terms of the GNU Lesser General Public License Version 2.1. The source code for this can be obtained from the following location: https://github.com/twall/jna/releases
- **Junit** which is distributed under the terms of the Eclipse Public License Version 1.0. The source code for Junit can be obtained from the following location: https://github.com/junit-team/junit/tree/master/src
- **Mimepull** 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://maven.java.net/content/repositories/releases/org/jvnet/mimepull/mimepull/
- **Objectweb ASM**, Copyright © 2000-2011 INRIA, France Telecom. All rights reserved.
- **Relique CSV Driver** which is licensed under the terms of the Lesser General Public License Version 2.0. This can be obtained from the following location: http://sourceforge.net/p/csvjdbc/code/ci/master/tree/
- **Rhino 1.6R7 and 1.7R2** which are licensed under the terms of the Mozilla License Version 1.1. The source code for this can be obtained from the following location: https://developer.mozilla.org/en-US/docs/RhinoDownload?redirect=no
- **Saxon** 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://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
- **Spring Framework** which is distributed under the terms of the Apache Software License Version 2.0. This product includes subcomponents with separate copyright notices and license terms.
- **Springsource JavaX Mail** 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: http://ebr.springsource.com/repository/app/bundle/version/detail?name=com.springsource.javax.mail&version=1.4.5&searchType=bundlesByName&searchQuery=mail
- **Springsource SLF4J 1.6.1**, Copyright © 2004-2008 QOS.ch. All rights reserved.
- **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/
- **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 License 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 Version 2.1. The source code for this can be obtained from the following location: http://www.7zsfx.info/files/7zsd_src_160_2712.7z

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.

This application also uses the following components which are distributed under the terms of the **Apache Software License Version 2.0**:

- Apache Ant
- Apache Axis
- Apache CFX
- Apache Commons Beanutils
- Apache Commons CLI
- Apache Commons Codec

- Apache Commons Collections
- Apache Commons Configuration
- Apache Commons DBCP
- Apache Commons Discovery
- Apache Commons FileUpload
- Apache Commons Imaging
- Apache Commons IO
- Apache Commons Lang
- Apache Commons Logging
- Apache Commons Net
- Apache Commons Pool
- Apache Commons VFS
- Apache Derby
- Apache Felix and dependencies
- Apache Geronimo
- Apache Jakarta HttpClient
- Apache Log4j
- Apache Neethi
- Apache OpenCMIS
- Apache POI
- Apache ServiceMix
- Apache Tomcat
- Apache WSS4J
- Apache Xalan
- Apache Xerces2 Java Parser
- Apache XMLGraphics
- Apache XML-RPC
- Barcode4j
- Google Collections
- Google GSON
- Jetty
- LMAX Disruptor
- OPS4J Pax Web
- org.json.simple
- Spring Dynamic Modules
- StAX
- XMLBeans

Eclipse Technology:

This Software includes unmodified Eclipse redistributables, which are available at www.eclipse.org. The Eclipse redistributables 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.

VSS Java FreeMarker:

This product includes software developed by the Visigoth Software Society (http://www.visigoths.org/).

This includes the following subcomponents that are licensed by the Apache Software Foundation under the Apache License, Version 2.0:

- freemarker/ext/jsp/web-app_2_2.dtd
- freemarker/ext/jsp/web-app_2_3.dtd
- freemarker/ext/jsp/web-app_2_4.xsd
- freemarker/ext/jsp/web-app_2_5.xsd
- freemarker/ext/jsp/web-jsptaglibrary_1_1.dtd
- freemarker/ext/jsp/web-jsptaglibrary_1_2.dtd
- freemarker/ext/jsp/web-jsptaglibrary_2_0.xsd
- freemarker/ext/jsp/web-jsptaglibrary_2_1.xsd

Java Advanced Imaging:

This application uses Java Advanced Imaging which is part of the Java SE framework and platform and which is distributed under the terms of the Oracle Binary Code License Agreement for the Java SE Platform Products and Java FX. Copyright 2013, Oracle America ,Inc. All rights reserved. Use is subject to license terms. ORACLE and JAVA trademarks and all ORACLE- and JAVA-related trademarks, service marks, logos and other brand designations are trademarks or registered trademarks of Oracle in the U.S. and other countries.

Further Components:

- This product includes software developed by the **JDOM Project** (http://www.jdom.org/).
- Portions of this software are copyright © 2010 **The FreeType Project** (www.freetype.org). All rights reserved.
- This product includes software developed by **JSON.org** (http://www.json.org/java/index.html).

Click to download the EULA as PDF