

OLTM Connect Send

Ad Hoc Mail Consolidation sample

Version 1.1

OL Connect Send Connect Send User Guide

2017-09-14

OBJECTIF LUNE Inc.

2030 Pie-IX, Suite 500

Montréal, QC, Canada, H1V 2C8

<http://learn.objectiflune.com>

© 2017 Objectif Lune Incorporated. All rights reserved.

Table of Contents

Introduction	6
Use of OL Connect Send	7
Basics	7
Processing steps of OL Connect Send	7
Dependence on browsers and display devices	8
Prerequisites	9
System requirements	10
Installation of the Connect Send sample	11
Downloading the application and the printer driver	11
Printer driver	11
Plugins	11
Downloading and configuring the sample files	11
Setting global variables and locations	12
Installing templates and configuration files	13
Deploy the Workflow configuration	13
Installation of the printer driver	13
Using the sample	16
Printing - sending documents	16
OL Connect templates in this solution	20
Letter: OLCS-Letter-tpl.OL-template	20
Web pages: OLCS-site-tpl.OL-template	20
"Waiting" web page	20
"Job" web page	21
"Summary" web page	22
"Dashboard" web page	22
"Oops" web page	22
OL Connect datamapper in this solution	23
OL Connect Send plug-ins in Workkflow	26
OL Connect Send Plugins	26
Configuring the OL Connect Send Workflow processes	26
Job transfer process	26
Interactive process	27
Production report process	27
Plugins	28
Job Processor plugin	28

Get Job Data plugin	32
Get Data plugin	37
Workflow processes in the sample	43
Startup process	43
Receiving data: OLCS_transfer	43
Processing / transmission process: OLCS_waiting	43
Waiting_webpage process	44
Waiting_processing_rest process	44
Displaying job information: OLCS_job	45
Job_webpage process	45
Job_preview_rest process	46
Job_boundaries_rest process	46
Completed transmission: OLCS_summary	46
Summary_webpage process	46
Summary_send_email process	48
Job overview process: OLCS_dashboard	48
Dashboard_web page process	49
Dashboard_rest process	49
Generate_output_rest process	49
Generate error page - OLCS_error_web page (subprocess)	49
Adjusting the sample	51
Duplex modes	51
Scenarios	52
Controlling the layout	53
Theme Switching	54
Adding a domain	54
Changing the colors	55
Adding a logo	56
Setting up a customer branded site	57
Envelope Handling	58
Envelope configuration file	59
Envelope drop-down	60
Envelope window area	60
Address extraction	62
Adding new envelopes	63
Dashboard buttons	63
Email notifications	64
How it works	65
Setting up email notification	66

Theming the email	67
Watermark	67
Custom watermark text	68
New features included in version 1.1	69
Legal Notices and Acknowledgments	71

Introduction

The aim of this document is to describe the basics of Connect Send and, at the same time, explain the functionality on the basis of an appropriate solution created and provided by Objectif Lune: the **AD HOC Mail Consolidation Sample**.

- You can watch the sample in action on demo.objectiflune.com. Under **Ad Hoc Mail Consolidation**, click **Demo** and follow the instructions. (If you have already installed the printer driver, you don't have to do that again.) Add a Connect Send printer with the given settings and print the provided Word file to that printer. The printer will trigger an interactive process on demo.objectiflune.com.

Installing the solution on your system will give you more insight in Connect Send.

Most of the properties of this solution can be adapted to specific, individual customer needs. The individual adaptations can be carried out by the customer – although it is advisable to have attended a Connect Basics Training Course in this case – or as a service provided by Objectif Lune Professional Services.

All important information can be found on our OL Connect Send page:
<http://help.objectiflune.com/en/#csend>.

Use of OL Connect Send

Basics

Connect Send offers the possibility to send print jobs to the automated Connect Workflow. This is done with the help of a **printer driver** that is installed with the user. A Postscript data stream is forwarded to OL Connect Workflow via this printer driver.

Further processing is then controlled individually via processes in **Connect Workflow**.

The requirement for setting up this OL Connect Send solution is the presence of OL Connect Workflow version 8.6 or later.

Processing steps of OL Connect Send

These are the 12 most important steps when printing with OL Connect Send, as seen from the client's perspective.

1. The user prints from any Windows application, using the OL Connect Send printer (driver).
2. The corresponding print job is received by the printer driver, compressed and sent to the Workflow Server – in several packets if necessary.
3. The packets are received in a Workflow server process with the HTTP Server input task and merged together via the Job Processor plugin.
4. The plugin communicates with the printer driver to ensure that the data have been transmitted in full.
5. The plugin assembles the packets to form a 7z archive and unpacks this to get the original job: a PostScript print file.
6. The plugin stores this print job in the location specified (in the plugin) under the defined name. (The definitions can use process variables.)
7. The plugin creates a metadata file for the print job with basic information.
8. In the event of a valid license, the plugin stores the data relevant to the job in the OL Connect Send database.
9. The plugin sends a status message back to the client's PC stating whether the transmission has been successful or failed.
10. The plugin reports the license status back to the printer driver.

11. In the event of a valid and activated license and an interaction process configured in Workflow, the default browser opens on the user's PC, initially with a web page from the process set during the printer installation. (As shown in this document, this could be a temporary page that forwards to an editing page as soon as the order has been sent in full.) The client can receive interactive feedback and define further processing via web pages.
12. The printer driver shows the status information for transmission in the Notification Area (also called 'systray').

Dependence on browsers and display devices

Initially, upon successful transmission (with licensed printing), the default browser opens on the client's PC. The full interactive process defined in Workflow however could involve other people using other devices.

When designing web pages (with the Connect Designer) for use in interactive processes it should be taken into account that the display and processing of HTML code can differ between browsers and that the way a web page is displayed also depends on the device on which it is viewed: a PC, tablet or smartphone.

Prerequisites

For this example the following is required:

- Connect v1.6 or higher
- Workflow v8.6 or higher
- A PReS Connect 8.6 or PlanetPress Connect 8.6 license with an OL Connect Send license.

System requirements

The printer driver is installed on the user's system running 32 and 64 bit versions of Windows .

Note

Windows 7: 32- & 64bit, Windows 8.1: 32- & 64bit, Windows 10 64bit , Windows Server 2008 R2 64bit, Windows Server 2012 R2 64bit.

The versions before Windows 7, Windows 8.0, Windows 2008/2012: 32bit or version before R2, Windows Server 2016 are not officially supported.

For the installation, the user must have the rights to install a driver. As communication between the printer driver and Workflow takes place via HTTP (or HTTPS), a corresponding connection between the printing PC and the OL Connect Workflow server must be usable.

Note

Due to certain properties of the printer driver, the OL Connect Send printer cannot be shared. It therefore has to be installed locally for the respective user. Terminal servers are not currently supported.

PDF display must be possible in the default browser.

Further information concerning system requirements can be found in the documentation at <http://help.objectiflune.com>.

Installation of the Connect Send sample

Downloading the application and the printer driver

Printer driver

You need the printer driver installation file for OL Connect Send. This can be found on our website <http://help.objectiflune.com/en/#csend-apps> in the Connect Send area.

In the ZIP file you will also find the Connect Send Printer Driver Installation guide (PDF) with all the relevant information concerning the installation of OL Connect Send.

Plugins

It is assumed you have installed the following components:

1. PlanetPress Connect v1.6 or higher, or PReS Connect v1.6 or higher.
2. Workflow v8.6 or higher.

The **plugins** for Connect Send in Workflow are installed automatically in Connect Version 1.6 (Workflow 8.6) and above.

Downloading and configuring the sample files

1. Download the [OLCS for AHM sample 1.1.zip](#) file to your local hard drive and unzip the folder.
 - a. The resulting folder will be your working directory for Connect Send and contains all configurations and sample files.
2. Move the folder to a location on your hard drive.
3. Locate the **OLCS-workflow.OL-workflow** file and open it in the Workflow 8 Configuration application

- a. The configuration is found in the following location: **OLCS_for_MHA_sample\configurations\Workflow\OLCS-workflow.OL-workflow**

Setting global variables and locations

1. Locate the **Global Variables** in the Tree View panel.
2. Double-click the **document root** variable and enter the path to the **workspace** folder of the solution (inside the **OLCS_for_AHM_sample** folder).
E.g. C:\Users\{your_user_name}\Documents\OLCS_for_AHM_sample\workspace
3. Set the path of the folder where the resources are located:
 - a. Click the **Workflow** icon in the upper left corner of the main application window.
 - b. Open the **Preferences** dialog of the Workflow 8 Configuration application.
 - c. Navigate to **HTTP Server Input 2**.
 - d. In the Resources field, enter the path to the **public** folder inside the workspace folder of the solution.
E.g. C:\Users\{your_user_name}\Documents\OLCS_for_AHM_sample\workspace\public.

Note

The solution assumes you are using the standard Connect Server settings. In case you've changed these settings (in the Connect Server Configuration tool) you will need to verify the following global variables:

1. rest_base_url
 - a. Default location: **http://localhost:9340/rest/serverengine**
2. rest_user
 - a. Default: **ol-admin**
3. rest_pwd
 - a. Default: **secret**

Make sure the same settings are stated in the preferences of the Workflow 8 Configuration application. To change these settings:

1. Click the **Workflow** icon in the upper left corner of the main application window.
2. Click **Preferences**.

3. Navigate to the **OL Connect** settings in the **Behavior** section.
4. Enter your Server Connection Settings.
5. Click OK.

Installing templates and configuration files

To add the templates and configurations to the Workflow environment:

1. Right-click the Connect Resources folder in the Tree View pane of the Workflow 8 Configuration application.
2. Choose **Import Content** from the contextual menu. The Import Connect Content dialog appears.
3. Navigate to the **Connect Resources** folder, located in the **configurations** folder of the solution.
E.g.: OLCS_for_AHM_sample\configurations\Connect Resources.
4. Select all files and click **Open**. The files are added to the Workflow environment.

Deploy the Workflow configuration

To deploy the OL Connect Send Workflow:

1. Make sure the **OLCS-workflow.OL-workflow** file is open. (It is found in the following location: OLCS_for_MHA_sample\configurations\Workflow\OLCS-workflow.OL-workflow.)
2. Use the **Send Configuration** option to deploy the Workflow configuration to the Watch service.
3. Start the Watch service (**Tools > Services Status > Start Watch**).

This runs the Startup Process and will initialize the OLCS_jobs database and envelope settings. This could also be achieved by running the Startup process in debug mode.

Installation of the printer driver

Follow the guidelines for installation of the printer driver outlined in the Connect Send Printer Driver Installation guide (PDF).

For the **Communication URLs** you can use the default settings.

- The URL for the data transfer is required for the HTTP connection with OL Connect Workflow (default: `http://localhost:8080/olcs_transfer`). In the present solution, this refers to the “OLCS_transfer” process, which accepts the relevant data.
If the Workflow server is active, it is useful to test the connection.
For greater security, HTTPS can be selected as the connection; it is possible to set whether only valid certificates are accepted by the Workflow server in this regard. HTTPS is not configured in this solution and can therefore not be used unless modified.
- If a license is valid for the local system and the interactive mode is activated, the action name of the HTTP Server Input task for interaction has to be given (default: “`olcs_interaction`”). In the present solution, this refers to the the “Waiting_web page” process that starts/executes the interaction.

OL Connect Send Printer Driver Setup

Communication URLs
Specify URLs for management of print jobs

OL Connect Send

URL for print job submission
`http://localhost:8080/olcs_transfer`

☒ Interactive mode ☐ Accept all certificates **Test**

HTTP action for interaction
`olcs_interaction`

< Back **Next >** **Cancel**

The **printer name** can be defined in the next step (default: OL Connect Send Printer #1), as well as the feedback level (Debug, Info, Warn, Error).

Note

The printer name cannot be changed afterwards.

A SysTray application for information is also installed with the installation of the printer driver. The **feedback level** of the OL Connect Send printer can be changed via “Modify” by running the installation again; other printers can also be added or removed. In the later Windows versions, printers with the same driver are shown under one icon in the “Devices & printers” view, though the individual printers are accessed via context menus.

Using the sample

A sample Word mail merge document is included in the **OLCS_for_AHM_sample_1.7.zip**.

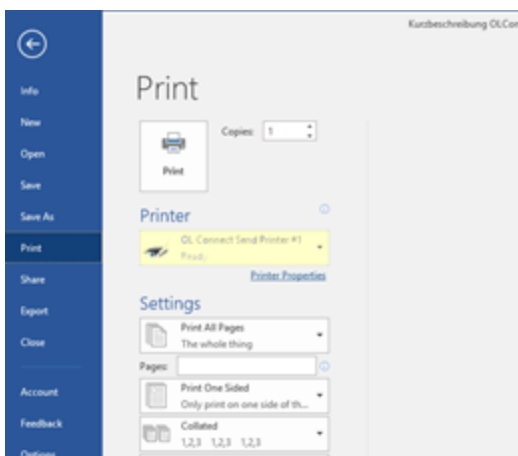
You can open the **mailing.docx** file from the **samples\MS Word sample** directory with Microsoft Word.

When prompted for the database to use select the **address-list.mbd** file located in the same directory.

Printing - sending documents

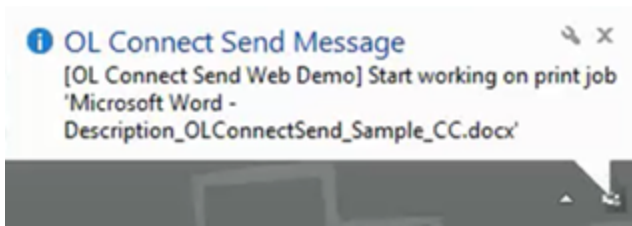
To print documents with Connect Send:

1. Click the Print button (or use the menu from the software used). The Print dialog appears.
2. In the Print dialog, select the OL Connect Send Printer and click OK.

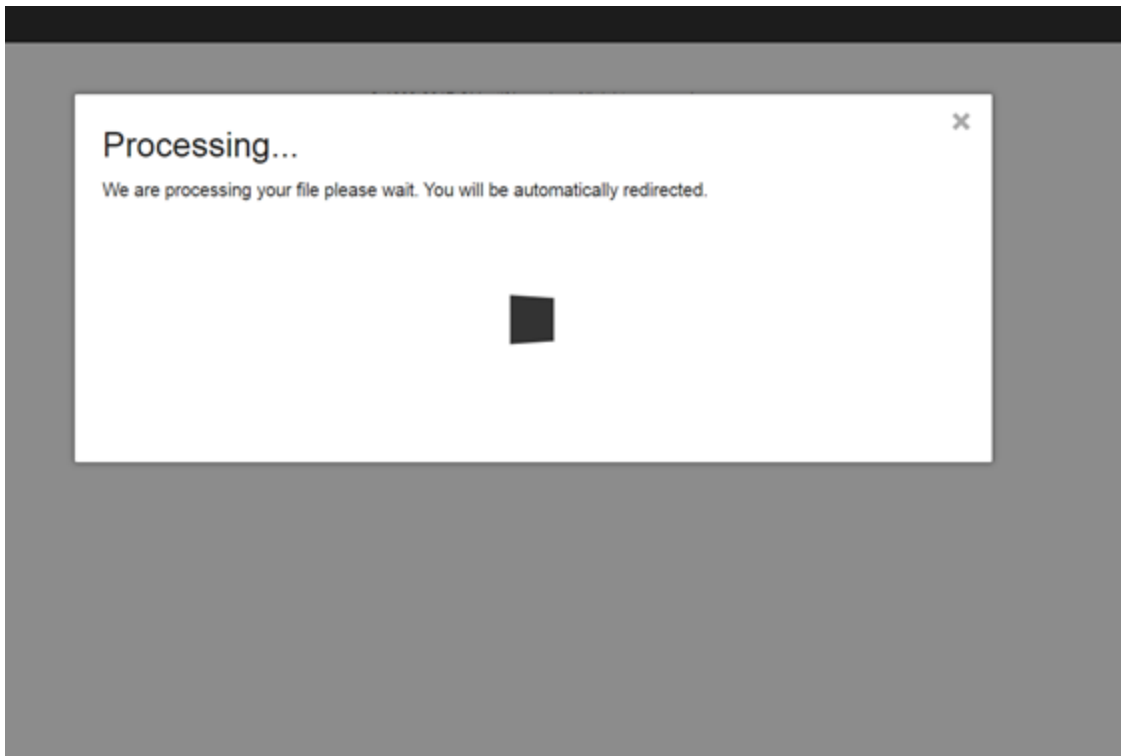


The OL Connect Send Printer generates a local PostScript file. This data is zipped and sent in one or multiple chunks to the Workflow server. Depending how the environment has been setup, information is sent using the HTTP or HTTPS protocol. As part of the process, the license is verified.

When there is no valid license, a notification is given in the bottom left area of the screen via the System Tray.



If the license is valid, the browser on the customer's machine is launched showing the web page as defined in the installation process of the printer driver. While the file is being processed the 'waiting' web page will be shown:



Once the file is transferred an interactive web page is displayed.

Job information

From	vandenhe
Reference	lvghmm0BIYZ
Document	Microsoft Word - mailing.docx
Pages	10 pages (5 document(s))

Boundary settings

Variable length

Note! Document boundaries are defined by the + character in the upper left corner of the pages.

Envelope

C5

Stationery First Page

Letterhead

Stationary Other Pages

None

Postage

1st Class

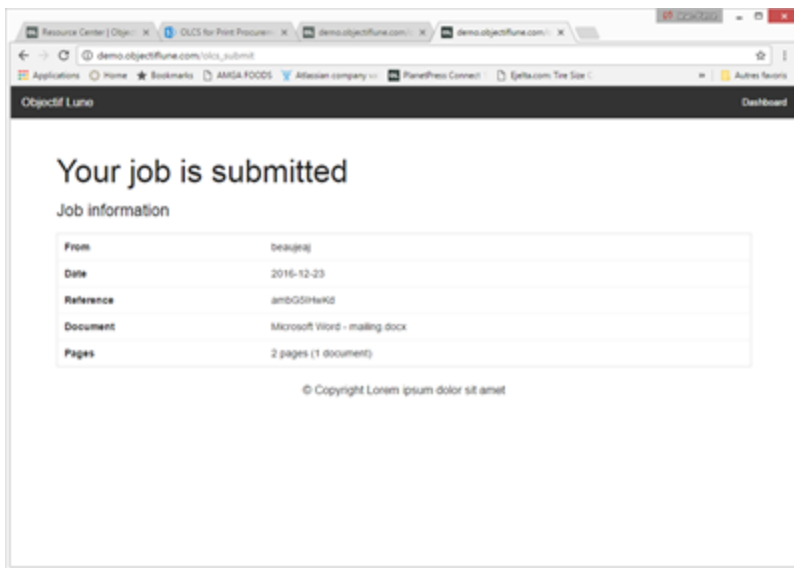
Cost Centre

610

A screenshot of a web-based letter creation interface for Business Bank. The interface is clean and modern, with a white background and a dark grey header. The Business Bank logo is in the top left. The letter content is centered. A placeholder address is shown in a green-bordered box. The letter body contains a salutation, a note about customizing the look, and instructions on how to change the design and logo. A 'Note!' section on the right side of the interface provides additional guidance. At the bottom, there is a navigation bar with 'Prev', a series of numbered buttons (1-5), and a 'Next' button. The current page is indicated by the number 1 being highlighted in the navigation bar.

Approve

Upon approval another web page summarizes the job that has been submitted.



OL Connect templates in this solution

Letter: OLCS-Letter-tpl.OL-template

This template is used as a window to the incoming print data. It contains elements that mimic the envelope windows and contains the media (and accompanying virtual stationery images). The template has the Letter page size.

When implementing a multi-tenant environment or to enable theme switching for domains one will typically create a variant of this template for each customer or even for different document types within organizations.

Web pages: OLCS-site-tpl.OL-template

This template contains all web pages in this solution that can be sent to the browser. It comes with a range of JavaScript files that handle the communication with the Workflow backend. An overview of the web page sections:

- **waiting**, the initial page shown the user
- **job**, the job submission page including the preview area
- **dashboard**, a basic dashboard page showing the approved and printed jobs
- **summary**, the job summary page shown after approving the job
- **oops**, a generic error page

Note

The OLCS-site-tpl.OL-template uses local JavaScript and CSS files for the Foundation framework. Remote resources are available but are currently unlinked. Using remote resources will improve the responsiveness (speed) of the web pages as the respective resources will be cached by the browser.

“Waiting” web page

The web page for waiting has been kept deliberately simple. Particularly important here are the functions from the integrated JavaScript **olcs-waiting.js** (and integrated functions from the

Foundation library).

After a short wait, redirection leads to the HTTP(S) action **olcs_process_Job** (see "Waiting_processing_rest process" on page 44) and therefore the display of the **Job** page for job processing.

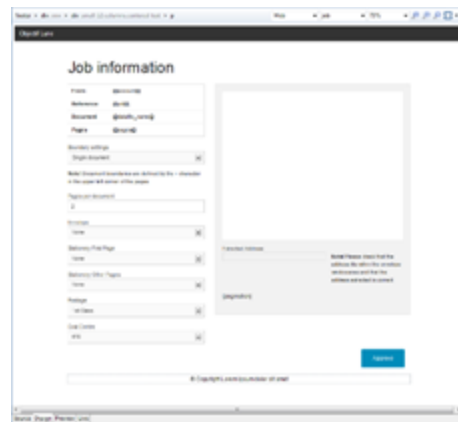
“Job” web page

The **Job** web page shows job information in the top left box regarding the sender, reference ID, document name and the total number of pages.

Selection of the type of boundary settings is then available to divide the complete job into one or multiple documents. The print stream can be defined as an individual document, a collection of documents with the identical number of pages or a collection of documents with a variable number of pages.

In the case of an identical number of pages, an additional input appears for the number of pages.

In the event of variable numbers of pages, a reference appears to the criterion of document separation. In this case separation occurs after a specific sign ("+") in a specific position on the first page of a document.



With the selection of the envelope, the associated address information can be displayed in the preview in the right-hand section of the page. The information about the relevant area is read from the file **envelopes.json**, in which the window positions for the envelope designations are stored.

In this solution, one can choose between the envelope types C5 and DL.

Different stationery can be specified for the first and following sheets in the documents; the options are Letterhead and Continuation.

The Postage and Cost Centre selection lists – bottom left – show options for billing specifications.

A preview of the currently selected document is displayed in the right-hand section of the page. Only one document is displayed at all times. It is updated in Workflow via the Create preview PDF plugin with the options chosen in the left-hand section and the document number (selected below the preview). The Extracted Address box displays the text found in the window area of the selected envelope type. Under the preview, a small text encourages the user to check whether the address is displayed correctly with the envelope type selected.

With the **Approve** button the form is submitted. This releases the job with the current settings for processing. The action of the Form is set to `"/olcs_submit"`, which triggers the HTTP Input action in the `Summary_webpage` process in Workflow. (With a relative URL the form data is sent to a different URL on the server.)

The `olcs_job.js` file is included in the **Job** web page.

"Summary" web page

For the purpose of confirmation, the **Summary** web page of the template is loaded, prepared with the transmitted data.

The Summary web page shows a summary of the job data. This page is opened when the Approve button has been pressed after job processing in the job page.



In this way, the licensed client receives a corresponding confirmation after completion of its print job.

"Dashboard" web page

The **Dashboard** web page lists the print jobs currently released and makes it possible to produce these via the Print button at the top. (To open this page in a browser, enter the following URL: `http://localhost:8080/olcs_overview` - or use one of the other HTTP actions defined in the `Dashboard_webpage` process in Workflow).

Displayed in the individual job lines are document properties (file name, number of pages and documents), document type, client's name, unique job ID, the internal index in Workflow as well as the creation date.

The functions from `olcs-dashboard.js` are used to create and update the Dashboard page.

"Oops" web page

The occurrence of errors has to be taken into account in real life. For the Workflow plugins it can be defined which process takes over in the event of errors.

In this solution, the error process is `OLCS_error_webpage`, which notifies the user of the occurrence of problems in the browser by displaying the **Oops** web page.



OL Connect datamapper in this solution

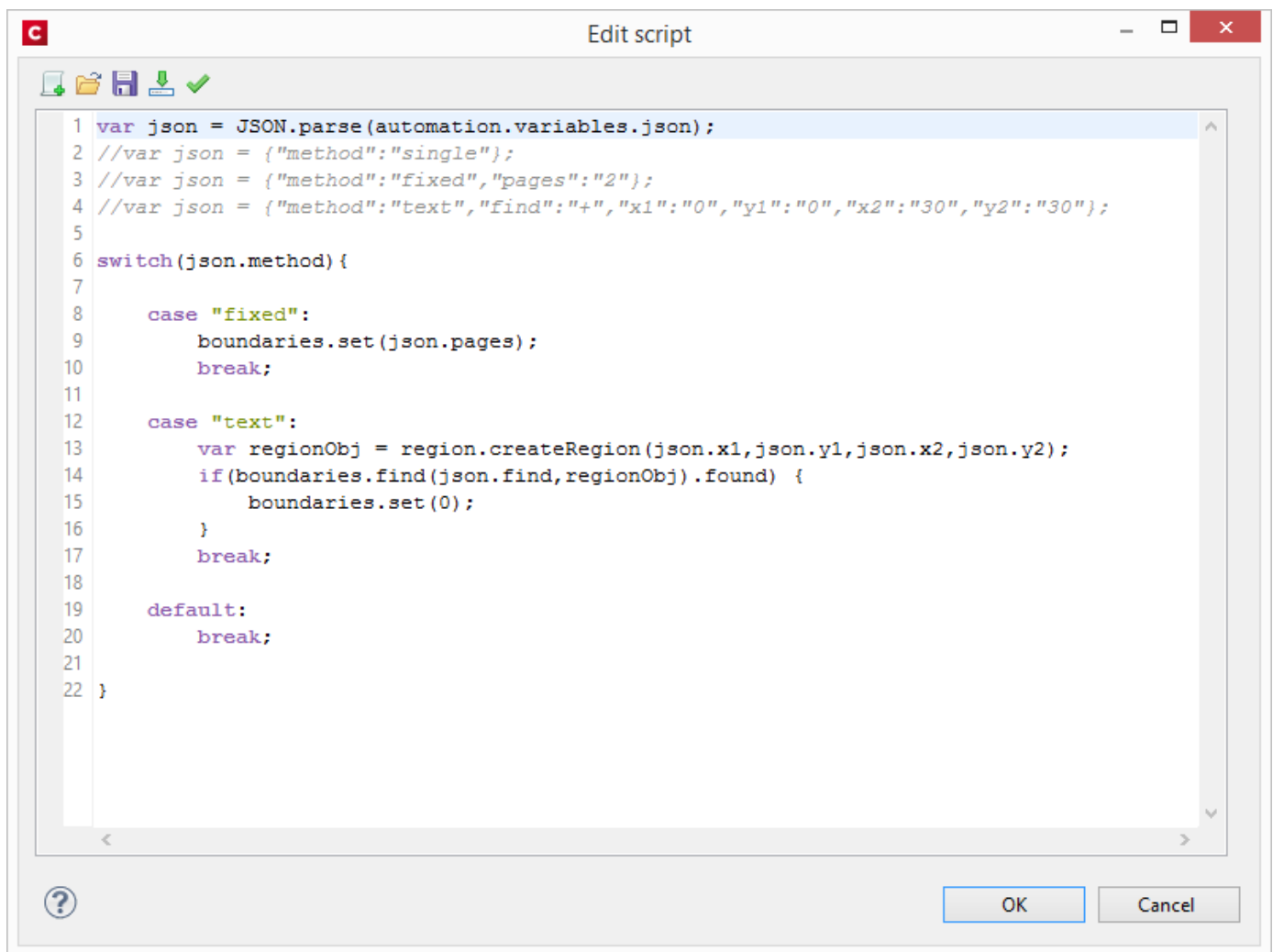
The incoming print job is converted to a PDF file. To set boundaries between documents in this PDF, the solution uses a data mapping configuration: **OLCS-define-boundaries-dm.OL-datamapper**.

The boundary detection method is handled via script. Before the boundaries can be set, the script needs to know how the PDF file should be regarded:

- as a single document, with all pages belonging together
- as a collection of documents, each with a fixed number of pages or
- as a collection of documents with a variable number of pages, in which each first page has a key text at a certain position. In this solution, a “+” sign is used in the top left corner of the page as a separator for dynamic document length. (In a future version of this sample this will be user definable.)

These parameters are provided by the "Job" web page and stored in the Workflow variable "JSON". This variable then contains the information in JSON format.

The script in the data mapping configuration reads this variable and sets the document boundaries accordingly:



- No boundaries are needed for an entire document (default).
- In the case of a fixed number of pages, a boundary is set after the given number of pages (json.pages).
- When a certain text marks the start of a new document – the “text” case –, this text is looked for in a certain area on the page – regionObj – and if it is found, a boundary is set at the start of that page.

Setting boundaries makes it possible to limit the preview to the document that is currently selected in the Job web page. The smaller the document, the faster the preview is generated of course.

Tip

When using MS Word you can put the marker (the "+") in the header of the first page of the document. Normally the header will be repeated for all pages. Check the link below to learn how to

create a different header for the first page: www.howtogeek.com.

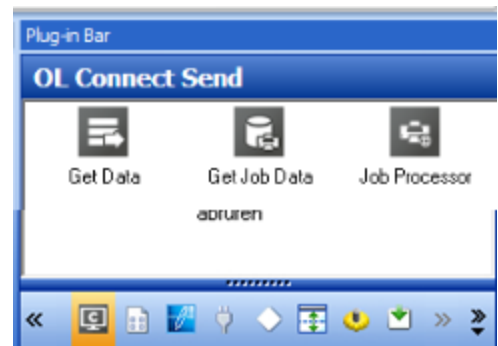
OL Connect Send plug-ins in Workflow

The **Ad Hoc Mail Consolidation** sample project may help you understand the Workflow processes for OL Connect Send and configure your own.

OL Connect Send Plugins

When installing OL Connect Workflow, an **OL Connect Send** group and, in this group, the three plugins **Get Data**, **Get Job Data** and **Job Processor** are inserted in the plugin tool list from version 8.6 on.

The plugin **Create Preview PDF** is provided in the **OL Connect** group in the plugin tool list.



Configuring the OL Connect Send Workflow processes

As can be seen during the installation of the OL Connect Send Printer Driver, OL Connect Send needs one Workflow process to handle the job transfer, and in licensed mode it needs at least one process to interact with the user once the server has received the job. Reports about the use of OL Connect Send might be produced in yet another Workflow process.

These Workflow processes and their related plugins are described below.

Job transfer process

The Workflow process that handles the **job transfer** is small and straightforward. It starts with an **HTTP Input task**. The action name of this HTTP Input task must match the last part of the URL for print job submission, set in the printer driver installer (by default: olcs_transfer). The **Job Processor** plugin is the only other task in this Workflow process.

For more information about the configuration of this process, see "Job Processor plugin" on page 28.

Interactive process

The core business of the **interaction** process(es) is: serving web pages to the customer and handling the customer's response, which comes down to changing (the settings for) the print job that has been received from that customer. As it has to be tailored to the situation in which it is used, several Workflow processes may be needed and they may be long and complex. It is therefore impossible to give a detailed instruction on how to configure the interactive process(es). It is however possible to name a few of the key components in such a process.

- Just as the job transfer process, the interactive process start with an **HTTP Server Input** plugin. The action name of this HTTP Input task must match the HTTP action for interaction given in the printer driver installer (by default: olcs_interaction). (For information about this plugin, see the Workflow Help: [HTTP Server Input](#).)
- Creating interactive processes for incoming print jobs using OL Connect Send requires that the relevant information about the respective job is available and can be used in Workflow. This is what the OL Connect Send **Get Job Data** plugin is made for. For more information, see "Get Job Data plugin" on page 32.
- Each web page served by an interaction process is generated by the **Create Web Content** plugin. It creates HTML output from a specified web template. That template needs to be created beforehand with Connect Designer and has to be sent to Workflow. (For information about this plugin, see the Workflow Help: [Create Web Content](#).)
- The **Create Preview PDF** plugin generates a PDF preview for a single record as fast as possible. It is typically used for previews embedded in web pages. As such it will often be used in an interactive process. (For information about this plugin, see the Workflow Help: [Create Preview PDF](#).)

Production report process

The key plugin in a Workflow process that produces reports about jobs received with OL Connect Send is the **Get Data plugin**. It allows to query the OL Connect Send database. For more information about this plugin, see "Get Data plugin" on page 37.

Plugins

Job Processor plugin

The **Job Processor** plugin appears in the **plugin Bar** area of **Workflow** under **OL Connect Send**.

The **Job Processor** plugin is an output plugin that must be added to a **Workflow** job transfer process that starts with an **HTTP Server Input**. The **Job Processor** plugin is the only other task in that process.

The action name of the HTTP Input task must match the last part of the URL for print job submission, that has to be set in the printer driver installer (by default: olcs_transfer).

After the job is processed, the HTTP Server Input returns a reply from the **Job Processor** plugin back to the **OLCS Printer Driver** in order to notify the user that the job has been received successfully (or failed if an error occurred).

Each incoming print job is checked against the license to determine if it can be handled in licensed mode or in unlicensed mode.

If OL Connect Send is **unlicensed**, the plugin stores the incoming job in the target folder using the specified file name, but it does not save any information in the database. The end user will receive a message in the Notification Area (also called "system tray") confirming the unlicensed status, and the printer driver will not request another web page.

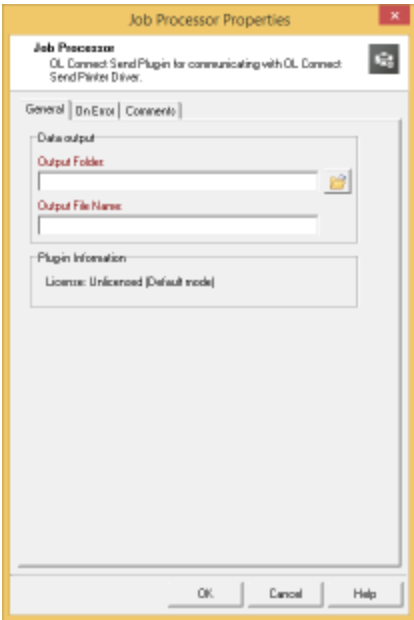
In **licensed** mode, the plugin will store all relevant information about each job in the OL Connect Send database. This database is a HSQLDB and is installed automatically. Subsequent Workflow processes can use the information in the database for additional processing (see "Get Job Data plugin" on page 32).

One job transfer process

It is strongly recommended that a single job transfer process for all Printer drivers is created, using the domain or machine's or user information to divert to any follow-up processes. This single transfer process can be set to "Self Replicating", so that parallelization is possible.

Configuring the plugin

To configure the **Job Processor** plugin, right-click it after adding it to the **Workflow** process, and click **Properties**.



The color of the property names in the **Job Processor Properties** dialog indicates that these fields can contain variables.

Workflow variables

To facilitate using job related information for the creation of the target folder and file name/s, the **Job Processor** plugin maps job relevant information to the standard **Workflow** variables (%1 to %8). The following mappings apply:

Information	Workflow Variable	When licensed	When unlicensed
Job ID	%1	Job ID	Job ID
License status for this job	%2	"Licensed"	"Unlicensed"
Username ¹	%3	The user name	"na"

Information	Workflow Variable	When licensed	When unlicensed
IP Address ¹	%4	The IP address	The IP address
No. of Pages ¹	%5	Number of pages of the job	Number of pages of the job
No. of Copies ¹	%6	Number of copies set by the user	Number of copies set by the user
Domain Name ¹	%7	The Domain Name	"na"
Machine Name ¹	%8	The Machine Name	"na"

1) These values originate from the Printer Driver machine.

The variables can and should be used to create dynamic file and folder names for each print job. This enables separating licensed and unlicensed jobs and/or storing the files by domain, machine and even user name.

File extension

If the output file name does not include an extension, the default extension “.ps” is added automatically.

Metadata

The plugin will create a **Workflow** metadata file containing basic information about the job. The values originate from the client machine.

In unlicensed mode, the user name, machine name and domain/workgroup name will not be available through the metadata.

Security

In order to provide security when printing over the internet, **OL Connect Send** includes several protective features.

HTTPS Communication

The Printer Driver can be set to use HTTPS for any job transfer. To do this **Workflow** must also be set to use HTTPS. This topic is covered briefly in the “Server URL Setting” section in this document (new Printer Driver installation), but full details can be found in the [Workflow](#) documentation.

Job Origin

Each print job will include unique information about the machine it has been sent from. This unique machine ID is calculated with a proven method and will be transferred, encrypted and enhanced. The enhancement will result in a different encrypted machine ID per print job, so that spoofing can be detected. On the server side, if spoofing is detected a respective message will be created.

Users can set up **Workflow** processes and filters to accept only specific jobs from known machine IDs, thus enhancing security.

Database connection

The **Job Processor** plugin works with a database to store all relevant job information. This database is a HyperSQL Database (HSQLDB, see https://en.wikipedia.org/wiki/HSQL_Database_Engine). It is installed as a service with the name **OL Connect Send DBServer** (the internal service name is **OLCSServer**).

Communication between the plugin and the database occurs using port 9001 (the default port for HSQLDB). However, there may be situations where this port is already blocked by another application and may need to be changed.

Several database settings can be modified by creating an ini file. This file must be named “OLCSSvc.ini” and stored in the same folder as the executable OLCSSvc.exe, located under “%CommonProgramFiles(x86)%\Objectif Lune\<Workflow Name>\Plugins\CPD”.

By adding the entry “DBPort = <new port number>” under [HSQLDBSETTINGS] and then restarting the service, the communication port is changed.

Note: **Workflow** has to be restarted after such a modification.

Timeout

During a job transfer from the **Printer Driver** to **Workflow**, a timeout could occur (indicated by a log entry like “ERROR: sendBinaryContents: Could not open request. Reason: 12002”). In this case, the timeout for the HTTP service in **Workflow** needs to be increased. We recommend using a value of more than 10 minutes (>600 seconds).

Additionally, the timeout in the browser on the client side should be enhanced. Please see the help pages for your browser about how to do this.

Compatibility

The **OLCS** components communicate with each other by using a well-defined and versioned protocol. As long as these components use the same protocol version, the job transfer will work even if the plugins themselves are changed. The protocol version can be found in the third part of the version number of the Printer Driver (i.e. in version number 1.2.5.98-17, the “5” indicates protocol version 1.5, omitting the leading 1). The Job Processor plugin shows the protocol version explicitly in the related frame.

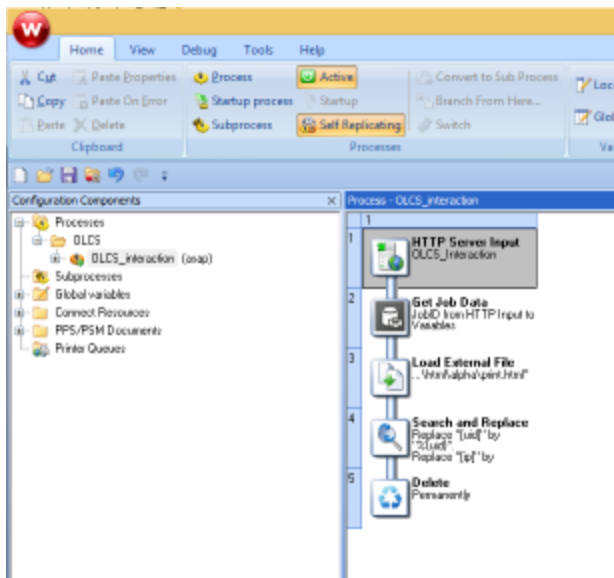
Any **Printer Driver** can communicate with any plugin, as long as this third version number part is identical.

Get Job Data plugin

Overview

Creating an interactive process for incoming print jobs using **OL Connect Send (OLCS)** requires that the relevant information about the respective job is available and can be used in **Workflow**. Job Information retrieval is made easy with the **Get Job Data** plugin.

The plugin appears in the **plugin Bar** area of **Workflow** under **OL Connect Send**.



Typically, it is used in the interaction process just after the initial **HTTP Server Input** plugin. The **Get Job Data** plugin gets all relevant information for the dedicated print job using the **Unique Job ID**.

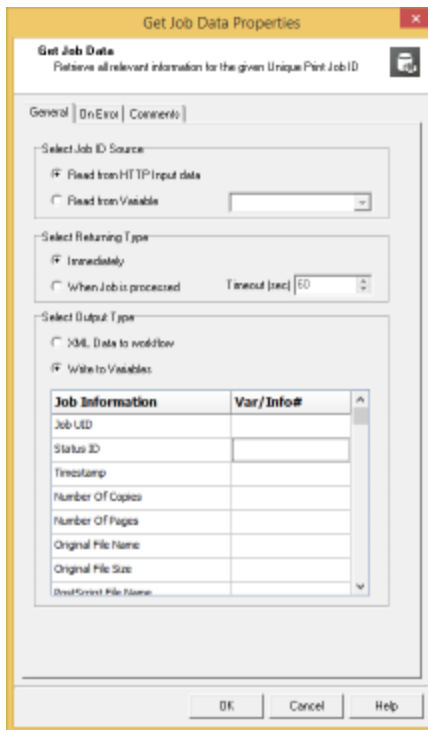
Unique Job ID: Whenever an **OLCS Printer Driver** is sending a print job to the **OLCS** plugin, it creates a unique ID string composed from 10 upper- and lowercase letters and digits e.g. "ri0zZdluLp". This Job ID is used in any communication between the **Printer Driver** and the plugin and is the leading identification element for this particular job. All job related information is stored in the underlying database and linked together by this Unique Job ID.

Note

The **Get Job Data** plugin gets data from the **OLCS** database which means it only works when **Connect** is in LICENSED mode.

The plugin settings

To configure the plugin, right-click it after adding it to the **Workflow** process, and click **Properties**.



General Tab

Three different settings affect the general behavior of the plugin:

- Where to get the **Print Job ID**.
- When to continue with the next step.
- Where to store the job information details.

Select Job ID Source

The plugin can be used in a generic way. Whenever information about a specific print job is required, it can be retrieved as long as the related job ID is known. However, the plugin has been implemented so that it can also be used in the “OLCS_Interaction” process. It can get the job ID from the incoming data of the **HTTP Server Input** plugin.

- **Read from HTTP Input Data.** When this option is selected, the plugin analyzes the incoming data and if it can find the job ID at the expected location, it uses it for further processing.

- **Read from Variable.** When selecting this option, any existing **Workflow** variable can be chosen via the drop-down field. In this case the plugin reads the Job ID from that variable.

Select Returning Type

Depending on this setting the plugin gets status information about the job before it has arrived or it gets information after the job has been completely received.

- **Immediately.** By choosing this option, the **Get Job Data** plugin will return as quickly as it can, providing it can find a matching job ID in the database. It is important to know that it will wait until information about the job is available. If no matching Job ID is found, the plugin will wait for 5 seconds and then raise an error, which can be acted upon using the **On Error** tab settings.

When selecting this option, the **Status ID** information has to be checked. A **Status ID** value of 1 or 6 indicates a fully processed job. Any value between 2 and 5 (inclusive) means that the job is still in progress.

- **When Job Is Processed.** When this option is selected, the plugin will query the database until the **Status ID** value is 1 or 6. If the status does not become 1 or 6 within the time defined via the **Timeout (sec)** input field, the plugin will raise an error (see above).

Select Output Type

- **XML Data to Workflow.** This will result in an XML file containing all job related data becoming the new **Workflow** job file. In this case, the incoming data file of the **HTTP Server Input** plugin is overwritten and lost.
- **Write to Variables.** This allows print job information to be stored in **Workflow** variables by using a simple drop-down list. In this case, the **HTTP Server Input** data will be kept as **Workflow** job file.

If the same **Workflow** variable is assigned more than once, the plugin will give a warning and will not close until the issue is fixed.

Returned Information

For each job received by the **OL Connect Send Job Processor** plugin the following values will be available.

Job UID: This is the 10 (ten) character long **Unique Job Identifier** string.

Date/Time stamp: This is the time when the matching job was initially created in the database. It is stored in UTC format plus time zone indicator inside the database. It will differ from the time stamp logged by the **Printer Driver** as well as by the **OLCS** main plugin.

Note

The **Printer Driver** machine time stamp in the **Printer Driver** log may significantly differ from this value.

Number of Copies: This is the value set by the **Printer Driver** for the number of copies (intended number of copies required for the print job). Some applications do not use the general print job information to define the number of copies. In such (rare) cases, the **Number of Copies** sent in the job can differ from what the user entered in the print dialog. For example: "IrfanView" does not use the regular **Copies** indicator, but instead sends the same job as many times as indicated by **Copies** in its print dialog.

Number of pages: This is the number of pages for one copy of the print job. This value is calculated by the Windows spooler, when processing the printing order. Please be aware that some applications do an implicit reformatting of jobs if the intended paper size does not match the paper size as selected in the print dialog. This may lead to the fact that the number of pages, as calculated by the spooler and reported by **OLCS**, can differ from that value as shown to the user in the application itself.

User name: This is the Windows user name of the user who started the application to produce the print job. It is not - in all cases - the user name of the user who is currently logged into the system.

Original file name: This is the "file name" as sent from the application to the Windows spooling system. It is taken from the name as it arrives in the spooler. Some applications add info to the name (like Notepad++) while others don't (like Adobe Reader). **OLCS** can only use what it gets from the spooler. It does not interact with the applications itself.

Original file size: The size of the print job - NOT the size of the document file.

Domain (workgroup) name: The name of the domain or workgroup the printing user belongs to. This is not necessarily the name of the domain the machine itself belongs to.

Domain / Workgroup Indicator: This domain name is the real domain name or only a workgroup name. For explanations about domains, domain names, users, user names, user domains, logged on users vs. application running users, machine names etc. please refer to the respective Windows help pages or ask your system administrator.

Machine name: The name of the machine the **OLCS** printer driver is running on as retrieved by the respective Windows API.

Machine GUID: The unique machine ID (see "Job Origin" on page 31) of the machine on which the job was produced. It can be used as an additional identification mark to validate the origin of the job.

On Error Tab

Please refer to the **Using the On Error tab** paragraph from the [Workflow](#) documentation.

Comments Tab

Please refer to the **Comments tab** paragraph from the [Workflow](#) documentation.

Get Data plugin

This plugin allows **OL Connect Send (OLCS)** users (admins, accounting personnel, print masters ...) to get information about all jobs received with **OLCS** on a dedicated machine. It allows queries of the **OLCS** Database to be made for the production of reports.

All job info that could be retrieved will be written to a temporary results file that is then passed on as the new **Workflow** job file. It can be used right after the **Get Job data** plugin in the same **Workflow** configuration. It could for example be saved using a **Send to Folder** plugin.



General Tab

Filter Options

Filters are required for:

- Start and end date (down to minutes).
- Domain(s)
- User(s)
- Machine name(s).

Except for start and end dates, it is possible to pass a list of multiple search criteria, separated with semicolons, containing:

- Workflow variables
- Job variables
- Names.

Note

No spaces are allowed around the listed names resp. before or after a semicolon.

Example 1

A valid user name search string would be:

%\{global.User};helen;%1;george napier

This would look for all entries, where the user name is:

- as currently stored in the global Workflow variable **User**
- "helen"
- as stored in the job variable number 1
- equals "george napier" (case insensitive).

The search criteria in this case is an **OR**.

If search criteria are entered in multiple input fields, all of them are combined with {{AND}}.

Example 2

Domain name is defined as {{objmtl.objectiflune.com}} and user name as {{rentel}}, then the result will only contain all the print job information for objmtl.objectiflune.com where the user name is "rentel".

Date and Time Definitions

Both date and time entries must be notated in UTC format. During runtime, the dates are checked and, if any other date/time notation is used, a Workflow error log entry is created.

UTC notation is described here: <https://www.cl.cam.ac.uk/~mgk25/iso-time.html>.

Valid date/time entries:

2016-07-12	It is possible to only define a date without a time
2016-%m-%d	Standard Workflow variables for year, month and day are allowed
2016-07-12 11:00	From and To dates may also have a time indicator (24 hour notation, separator from the date is a space character, separator between hour and minute is a colon)

It is possible to define the same date for **From Date/Time** as for **To Date/Time**. However, entering the same info (without time information) would lead to getting no entries.

Operators

- Searches are case-insensitive.
- Searches with multiple entries in one criteria: OR.
- Searches with entries in different criteria: AND.

Results

For each job that matches the search criteria, the following information will be put in the resulting data file:

- Job UID
- Date/Time stamp
- Number of copies
- Number of pages
- User name
- Original file name
- Original file size
- Domain (workgroup) name
- Domain / Workgroup Indicator
- Machine name
- Machine GUID.

Results File Format

The following result file formats are selectable:

- XML
- JSON
- CSV (Separator = semicolon (0x3B), string indicator = quote (0x22)).

Note

This file is not automatically saved to disk. The retrieved job info is written to a temporary results file that will be passed on as the new **Workflow** job file. To save the results file, use a **Send to Folder** plugin and configure that appropriately.

Note

Please keep in mind that the **Get Job Data** plugin gets data from the **OLCS** database which means it only works when **Connect** is in LICENSED mode.

Returned Information

For each job received by the main OLCS plugin the following values will be available.

- **Job UID:** This is the 10 (ten) character long **Unique Job Identifier** string.
- **Date/Time stamp:** This is the time when the matching job was initially created in the database. It is stored in UTC format plus time zone indicator inside the database. It will differ from the time stamp logged by the **Printer Driver** as well as by the **OLCS** main plugin.

Note

The **Printer Driver** machine time stamp in the **Printer Driver** log may significantly differ from this value.

- **Number of Copies:** This is the value set by the **Printer Driver** for the number of copies (intended number of copies required for the print job). Some applications do not use the general print job information to define the number of copies. In such (rare) cases, the **Number of Copies** sent in the job can differ from what the user entered in the print dialog. For example: "IrfanView" does not use the regular **Copies** indicator, but instead sends the same job as many times as indicated by **Copies** in its print dialog.
- **Number of pages:** This is the number of pages for one copy of the print job. This value is calculated by the Windows spooler, when processing the printing order. Please be aware

that some applications do an implicit reformatting of jobs if the intended paper size does not match the paper size as selected in the print dialog. This may lead to the fact that the number of pages, as calculated by the spooler and reported by **OLCS**, can differ from that value as shown to the user in the application itself.

- **Username:** This is the Windows user name of the user who started the application to produce the print job. It is not - in all cases - the username of the user who is currently logged into the system.
- **Original filename:** This is the "file name" as sent from the application to the Windows spooling system. It is taken from the name as it arrives in the spooler. Some applications add info to the name (like Notepad++) while others don't (like Adobe Reader). **OLCS** can only use what it gets from the spooler. It does not interact with the applications itself.
- **Original file size:** The size of the print job - NOT the size of the document file.
- **Domain (workgroup) name:** The name of the domain or workgroup the printing user belongs to. This is not necessarily the name of the domain the machine itself belongs to.
- **Domain / Workgroup Indicator:** This domain name is the real domain name or only a workgroup name. For explanations about domains, domain names, users, user names, user domains, logged on users vs. application running users, machine names etc. please refer to the respective Windows help pages or ask your system administrator.
- **Machine name:** The name of the machine the **Printer Driver** is running on as retrieved by the respective Windows API.
- **Machine GUID:** The unique machine ID (see "Job Origin" on page 31) of the machine on which the job was produced. It can be used as an additional identification mark to validate the origin of the job.

On Error Tab

Please refer to the **Using the On Error tab** paragraph from the [Workflow](#) documentation.

Comments Tab

Please refer to the **Comments tab** paragraph from the [Workflow](#) documentation.

Workflow processes in the sample

This description of the Workflow processes in the sample also explains how they work together with Connect templates (see "OL Connect templates in this solution" on page 20) and a data mapping configuration ("OL Connect datamapper in this solution" on page 23).

Startup process

The startup process is executed once at the start of the service to perform initialization.

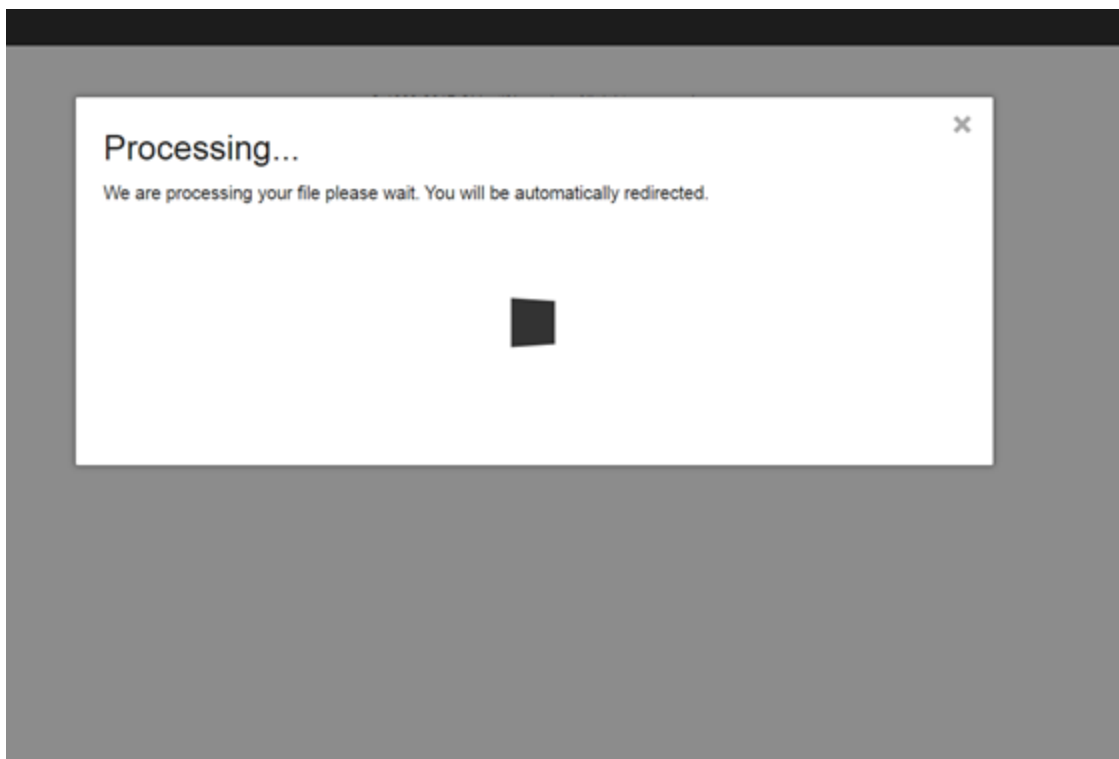
- It loads the data of the **envelopes.json** configuration file into Workflow with the envelope data (1st branch, global variable **params_envelopes**).
- It creates a group in the Data Repository for **OLCS_jobs** (2nd branch).
- It creates the string for authentication formed from the user name (**rest_user**; default: ol-admin, adjust where applicable) and the password (**rest_pwd**; default: secret, adjust where applicable) in the global field **rest_auth_string**.

Receiving data: OLCS_transfer

The OLCS_transfer process receives the data coming from the printer driver and stores them in the file system.

Processing / transmission process: OLCS_waiting

In this process, the **Waiting** web page is created (from the connect template **OLCS-site-tpI**) which is sent back to the browser and informs the user that the job is currently being processed.



The following two processes run during this display.

Waiting_webpage process

The **Waiting_webpage** process is triggered via the HTTP action **olcs_interaction**. The HTTP action for interaction can be set during the installation of the printer driver. **Olcs_interaction** is the default feedback action used in the printer driver installation.

After updating Workflow variable %9 with the user ID from the transmission of the printer driver, the web section **Waiting** from the Connect template **OLCS-site-tpl** is displayed.

Waiting_processing_rest process

This process is triggered via the HTTP action “**olcs_process_Job**” from the HTML page displayed here and first retrieves the job data. A condition ensures that processing only takes place if the data are not empty. If this is the case, the PostScript print file received by the printer driver is converted and stored in the job directory under a unique name.

The associated job information is then added to the **OLCS_jobs** group in the Data Repository: user account, pages, documents, record set ID, file name, unique ID, status (=‘mapped’), parameters, paper type and domain name.

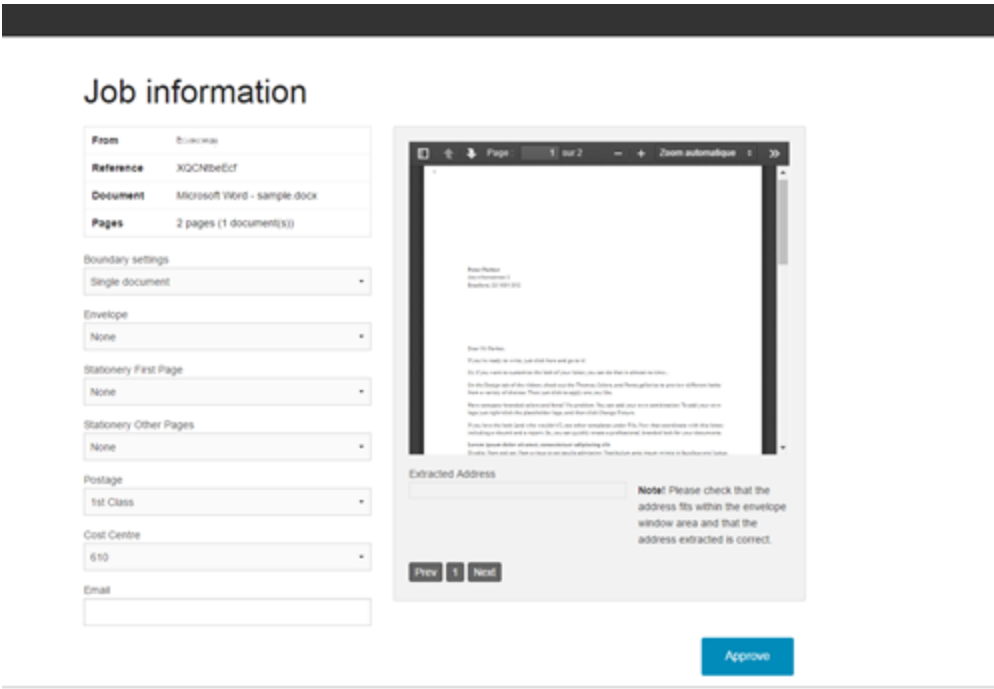
A script then provides the basis for limited further processing.

Displaying job information: OLCS_job

This process is started after successful processing from the processing/transmission process.

In this process, the **Job** web page is created (from the Connect template **OLCS-site-tpl**) and sent back to the browser.

This HTML view is sent to the browser:



Job_webpage process

The HTTP action **olcs_job** starts the processing of the first web page display, with the web section **Job** from the **OLCS-site-tpl** template being returned in the event of error-free

processing.

An error is issued if the record requested via the unique ID in the Data Repository in the group **OLCS_jobs** is not found or does not have the “mapped” status. The mapped status is set in the **Waiting_processing_rest** process when the file is processed without errors and is stored in the system.

Job_preview_rest process

After a user action in the web page, the document preview is updated in the browser via the HTTP action **olcs_preview**.

The data set from the Repository is first read from the input data. In the job, a PDF is created for the document that is currently selected and stored in the public\previews registry with a unique ID. The address information is also identified according to the envelope selected.

Job_boundaries_rest process

From the web action **olcs_updateboundaries**, the preparation of the document data is renewed based on the current differentiations or changes made on the web page. This is used to display form letters in this solution.

First of all, the unique ID is generated from the transfer data, the corresponding PDF is loaded and the data for the **OLCS-define_boundaries-dm** data mapping (see "OL Connect datamapper in this solution" on page 23) created from this. Via script, the page offset of the respective document in the PDF is then determined from the length of the individual documents, added in the metadata and the data sets updated in the database with this.

The number of documents and the data set ID for the current job are updated in the Data Repository.

Completed transmission: OLCS_summary

Summary_webpage process

This process is started as soon as the **Approve** button has been clicked (thus submitting the form).

In this process, the **Summary** web page (from the Connect template **OLCS-site-tpl**) is generated and sent back to the browser.

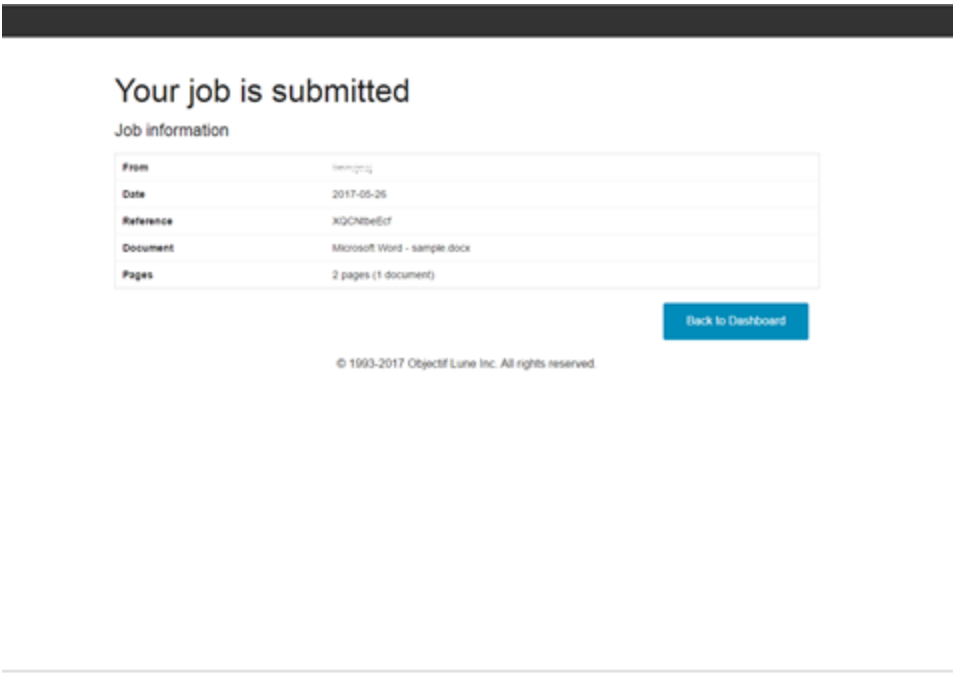
It is first checked in the **OLCS_jobs** group in the Data Repository whether the job can be found and has the 'mapped' status. If both are true, a JSON string with the parameters is created from the job file (the request XML) and stored in the Jobs directory in the assigned subdirectory as *.json; the status of the job is set to approved in the Data Repository.

Tip

To simplify this process, a new plugin: the XML/JSON Conversion task will be available as of Connect Workflow 8.8.

The request XML is written to the Email folder.

Finally the job data are read from the Data Repository and displayed in the Summary web section of the **OLCS-site-tpl** template in the browser.



Summary_send_email process

This process starts as soon as the Summary_webpage process has sent the request XML to the Email folder.

It outputs an email summarizing the job data if a valid email address has been provided.

Job overview process: OLCS_dashboard

In this process, the **Dashboard** web page is generated (from the Connect template **OLCS-site-tpl**) and sent back to the browser as soon as the action **olcs_overview** is triggered in the browser: **http://127.0.0.1:8080/olcs_overview**.

Here, the available, released data from the previous processes are displayed in a table and can be released for printing. As soon as data are printed the colour of the dot before the document changes from orange to green.

This HTML view is sent to the browser:

Job overview

Show 15 entries

Search: sample_letter.docx

Document	Type	Account	UID	ID	Created
<div><div></div>Microsoft Word - sample_letter.docx 2 pages (1 document)</div>	Letter	web user	bCSAdYMK8d	290	2017-05-11
<div><div></div>Microsoft Word - sample_letter.docx 2 pages (1 document)</div>	Letter	web user	mc77Tt24ar	288	2017-05-10
<div><div></div>Microsoft Word - sample_letter.docx 2 pages (1 document)</div>	Letter	web user	CPMHov4nAO	286	2017-05-10
<div><div></div>Microsoft Word - sample_letter.docx 2 pages (1 document)</div>	Letter	web user	RqYSlgZY8	285	2017-05-10
<div><div></div>Microsoft Word - sample_letter.docx 2 pages (1 document)</div>	Letter	web user	1rZG3ABqW	284	2017-05-10
<div><div></div>Microsoft Word - sample_letter.docx 2 pages (1 document)</div>	Letter	web user	P3iHfMowEB	279	2017-05-09
<div><div></div>Microsoft Word - sample_letter.docx 2 pages (1 document)</div>	Letter	web user	meJ8ww2Q4i	278	2017-05-09
<div><div></div>Microsoft Word - sample_letter.docx 2 pages (1 document)</div>	Letter	web user	CMERHyCKQ8	277	2017-05-09
<div><div></div>Microsoft Word - sample_letter.docx 2 pages (1 document)</div>	Letter	web user	SkaF077KqL	276	2017-05-09
<div><div></div>Microsoft Word - sample_letter.docx 2 pages (1 document)</div>	Letter	web user	GP5GesFuGa	275	2017-05-09

Showing 1 to 10 of 10 entries (filtered from 152 total entries)

Previous

1

Next

Approved

Printed

© 1993-2017 Objectif Lune Inc. All rights reserved.

Page 48

OL

Dashboard_web page process

In this process, the web section **Dashboard** is prepared from the data of the HTTP requests of `olcs_overview`, `olcs_dashboard` und `olcs_dash`. In this away, the user obtains an overview of the current jobs.

Dashboard_rest process

Through the HTTP action **olcs_data** the information from the `OLCS_jobs` group in the Data Repository are read and written into the Workflow variable `%9` as a JSON string. This JSON string is returned to the caller as the current job.

Generate_output_rest process

This process handles the output of the print job via Workflow from the HTTP action **olcs_output**.

Using the unique identifier provided, the data from the `OLCS_jobs` group of the Data Repository are retrieved and, with it, the data from the Connect database. The mappings of the stationery from the repository are then updated in the data.

After the steps **Create printout** (using the existing PDFs), **Create print job** (OLCS-job-creation.OL-jobpreset) and **Create output** (OLCS-Generic-PDF.OL-outputpreset), the output file is stored in the Output directory.

In the Data Repository the status of the order is then set to completed.

Generate error page - OLCS_error_web page (subprocess)

In the event of errors in the aforementioned processes, this subprocess is launched. It sends an error page based on the **Oops** web section of **OLCS-site-tpl**.

This HTML view is sent to the browser:

Oops!

We can't seem to find the page you're looking for. Please try again later. If you think this message is wrong, please contact your system administrators.

©2017 Objectif Lune

Adjusting the sample

The following sections provide detailed information about a number of features in the sample, especially features that weren't available yet in the 1.0 version of the sample (see "New features included in version 1.1" on page 69). Understanding how these features are implemented will help you adjust the sample to your needs.

Duplex modes

The template used to render the previews (OLCS-A4-tpl.OL-template or OLCS-Letter-tpl.OL-template) now supports the following duplex modes:

- Simplex
- Duplex no media/stationery specified
- Duplex

These modes are triggered by the Double Sided and Stationery drop-downs in the Job Information web page. Below a description of these modes.

Job information

From	vandenhe
Reference	JSDFePYSzD
Document	Microsoft Word - mailing.docx
Pages	10 pages (5 document(s))

Boundary settings

Fixed length

Pages per document

2

Envelope

None

Stationery First Page

Letterhead

Stationery Other Pages

Continuation

Double Sided

Yes

Postage

4 of Place

Page: 1 of 4 25%

Extracted Address

Note! Please check that the address fits within the envelope window area and that the address extracted is correct.

Prev 1 2 3 4 5 Next

Scenarios

Simplex

Each printed page is outputted on the front side of a sheet. Reverse sides are not shown in the preview.

Number of pages printed for a recipient (e.g. document): 3

Output:

- Page 1: content
- Page 2: content
- Page 3: content

Duplex no media/stationery specified

Each printed page will be placed on the front or reverse side of a sheet, starting on the front side of sheet 1. Printing an odd number of pages results in a blank reverse side of the last sheet.

Number of pages printed for a recipient (e.g. document): 3

Output:

- Sheet 1: Front: content
- Sheet 1: Reverse: content
- Sheet 2: Front: content
- Sheet 2: Reverse: -

Duplex with media/stationery specified

In this scenario the first sheet acts as the company letterhead with the terms and conditions on the reverse side and therefore it only accepts content on the front side of the first sheet. The remaining pages are placed on the front and reverse sides of the subsequent sheets.

Number of pages printed for a recipient (e.g. document): 3

Output:

- Sheet 1: Front: content (letterhead)
- Sheet 1: Reverse: -
- Sheet 2: Front: content (continuation)
- Sheet 2: Reverse: content

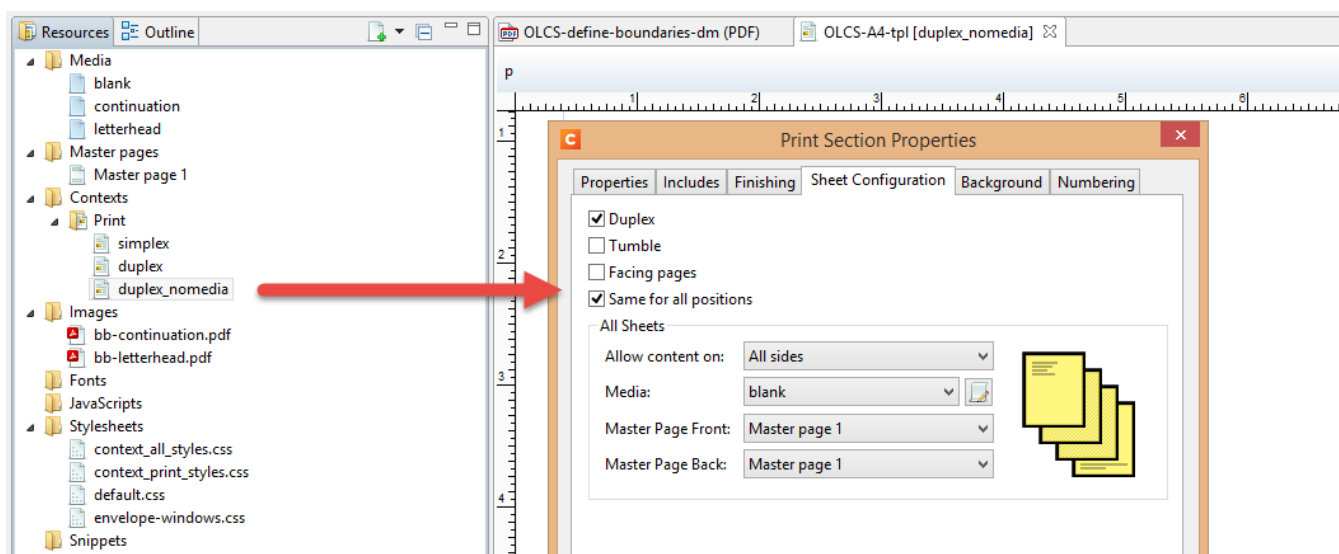
Number of pages printed for a recipient (e.g. document): 4

Output:

- Sheet 1: Front: content (letterhead)
- Sheet 1: Reverse: -
- Sheet 2: Front: content (continuation)
- Sheet 2: Reverse: content
- Sheet 3: Front: content (continuation)
- Sheet 3: Reverse: -

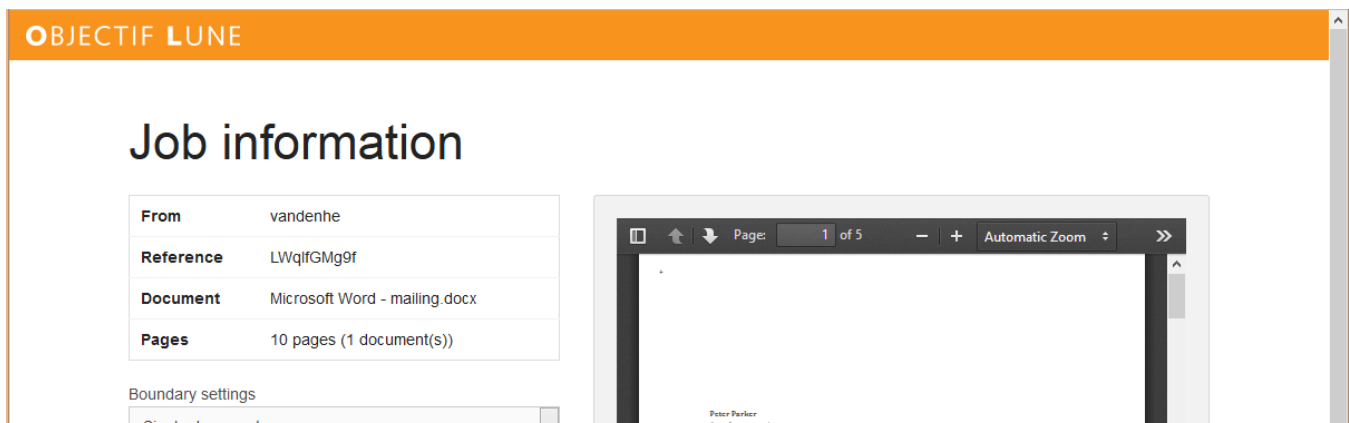
Controlling the layout

toggling the duplex modes is handled by enabling/disabling print sections from within a control script in the preview template. The preview template contains a print section for each mode. Each section has the required Duplex and Allow content on options set in the Sheet Configuration dialog.



Theme Switching

Showing a domain specific theme can be implemented in various ways. One could create a web template per domain and add checks in the Workflow processes to dynamically load the template file for that domain. Alternatively one could use the same template file but dynamically load a stylesheet file. The latter is implemented in this version of the OLCS-site.tpl file of the Ad Hoc Mail demo. This approach allows to change the color scheme of the pages dynamically and add a logo image for example, without changing the actual template files.



The visitor-facing webpages (waiting, job, summary and error/oops pages) dynamically load a CSS stylesheet file based on the domain of the visitor/user. The domain is passed by the printer driver as part of the print job and received by Workflow upon showing the first web page.

The domain name is passed to the template via the Job Info %8 variable in Workflow. The dynamic CSS script in the OLCS-site-tpl template reads this information and adds a reference to a domain specific stylesheet file (styles.css). This file is served by Workflow and located in a domain specific CSS folder in the workspace folder (aka the Resources or iRes folder). The stylesheet file contains a subset of the Foundation style rules, overwriting the color of the top header and the color of the buttons.

The styles.css file is typically found in the following location:

```
C:\Users\{your_user_name}\Documents\OLCS_for_PPA_sample\workspace\public\domains\{domain_name}\css\
```

Adding a domain

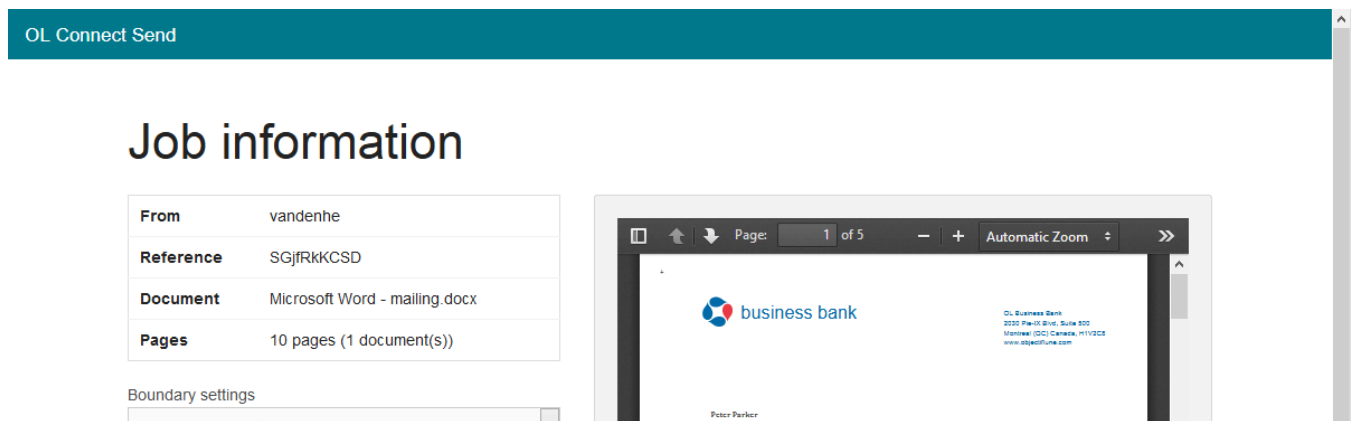
To add a theme for a domain:

1. Locate the domains folder. This folder is typically found in the following location:
C:\Users\{your_user_name}\Documents\OLCS_for_PPA_
sample\workspace\public\domains\
2. Create a copy of the DEMO folder and rename this to the domain of the visitor.
The domain name can be retrieved from the System dialog on the machine of the customer/user. Simply right-click This PC and choose Properties.

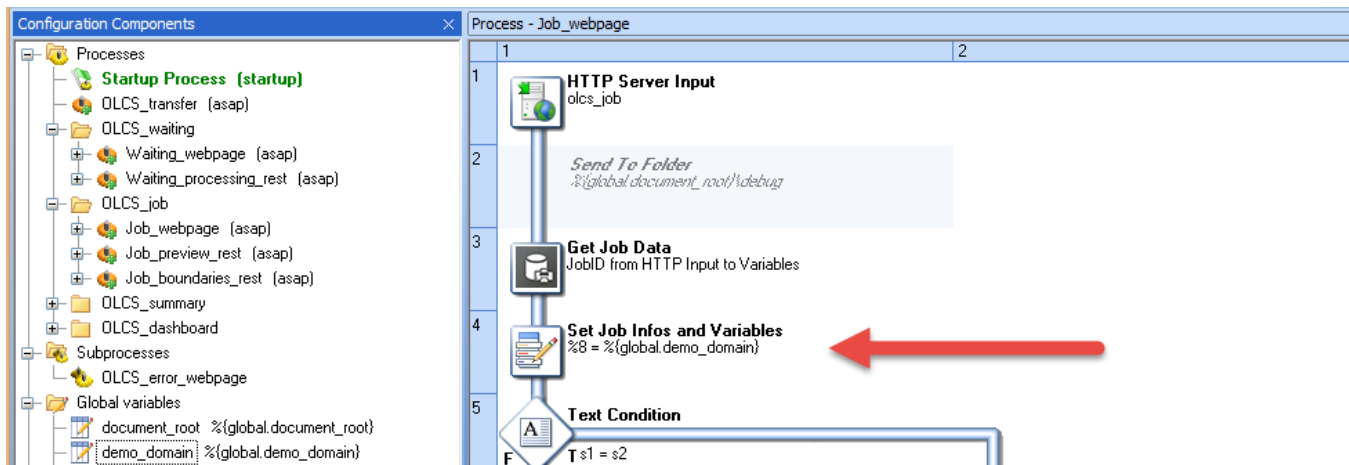
Changing the colors

To change the colors for the domain:

1. Locate the **styles.css** file in the CSS folder of the new domain.
2. Open this file in a text editor (for example NotePad++). The stylesheet file contains various color overwrites for the top bar and buttons.
3. Change the color values and save the stylesheet file.
4. Print a document from the respective domain to view the result.



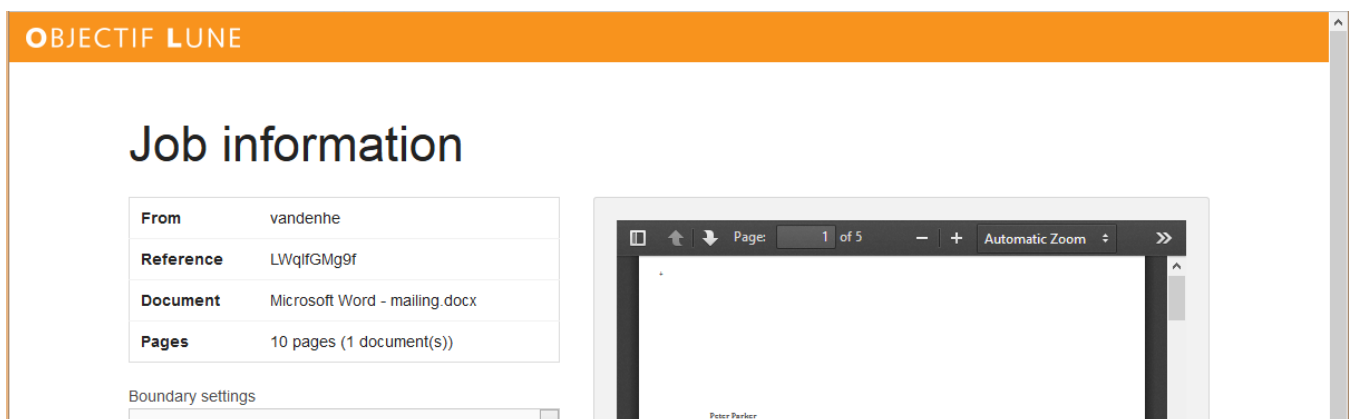
While implementing the Workflow processes one could add a Set Job Infos and Variables plugin and write a hard coded domain name to Job Info %8. The plugin should be placed after the Get Job Data step as this plugin populates Job Info 8 with the domain name retrieved from the Connect Send Job database.



For convenience, we added the Set Job Infos and Variables steps described above to the sample processes but made them inactive. Right-click the respective step in the process and choose **Ignore** from the contextual menu to enable or disable them. The domain name is taken from a global variable called **demo_domain**.

Adding a logo

There are various techniques to add a logo to the top bar. One can add an `` element to the respective location in the HTML or set the image as a background image. The logo as background image can be controlled from within the dynamically linked stylesheet and doesn't require changes to the OLCS-site-tpl template file.



To add a logo:

1. Create a logo in an image application or download one from the internet. Make sure it is saved in .png format. The .png format allows you to use a transparent background which will make it blend nicely into the colored bar. Ideally the logo has a landscape orientation.
2. Save the logo to the Images folder in the folder of the respective domain.
3. The following rules in the styles.css file hide the default text in the upper left corner of the top bar and add the logo as background image using the **:before** pseudo selector.

```
.top-bar h1.title a:before {content: ''; display: inline-block; visibility: visible; background-image: url ('../images/logo.png'); background-repeat: no-repeat; background-size: contain; height: 18px; width: 300px; vertical-align: middle; margin-top: -2px; /* nudge the logo to the correct position*/}
.top-bar h1.title a{visibility: hidden;}
```

Setting up a customer branded site

When doing a demo for a prospect it is not likely that you are printing from their domain. The following steps explain how to set up a demo branded for a customer:

1. Locate the domain folder in the workspace directory.
2. Duplicate the DEMO folder and rename the folder to the name or domain of the customer.
3. Visit the website of the prospect and use a color picker or source inspector to retrieve the primary and secondary color of the site/the corporate style.
Color pickers are available as browser plugins. Alternatively create a screenshot and use GIMP or Photoshop to retrieve the color.
4. Open the styles.css file of the new domain folder in a text editor (e.g. NotePad++) and replace the various color values with the ones retrieved from their site.
5. Download a .png version of the customer's logo. Typically this can be taken from their website. Alternatively google for: **{customer_name} logo png** and check the **Images** option in the search result.
6. Put the logo in the Images folder located in the DEMO folder and rename it to: **logo.png**.
7. Launch Connect Workflow and open the OLCS workflow configuration.

8. Enable the **Set Job Infos and Variables** step which is placed after the Get Job Data step.
9. Locate the **Global variables** section in the Configuration Components panel and double click the **demo_domain** entry.
10. Enter the name matching the new domain folder.

Save the configuration and click the Send Workflow icon to deploy it to the Workflow server.

Envelope Handling

The Job Information page of the Ad Hoc Mail sample allows the user to select an envelope for the print job. Selecting an envelope invokes an update of the PDF preview area. In this process the window area is drawn on the first page of the preview and the text in the area is extracted. The extracted text is shown below the PDF preview. These actions allow the user to validate if the address isn't clipped by the window and shows the text that is used when doing postal sorting (the latter is not covered by the Ad Hoc Mail sample).

In addition the solution checks if the number of sheets produced will fit in the envelope. When it doesn't fit an error message will be shown asking the user to change the job options or select a different envelope type.

All of these features are driven by a centrally stored configuration file containing the list of envelope types with their specifications. The following sections describe how these features are implemented.

Job information

From	vandenhe
Reference	ahjclxd6E
Document	letter-business bank-1k.pdf
Pages	2000 pages (1000 document(s))

Boundary settings

Fixed length

Pages per document

2

Envelope

C5

None

C5

DL

Lettermeau

Stationery Other Pages

Continuation

Double Sided

Yes

Postage

1st Class

business bank

Sarah Kunz
Manning Engine Repair Group
212, 92th Street
P.O. Box 10481
N3F 4G7 Manning

Date
Apr 19, 2017

Subject
Important information about your business current accounts

Dear Sarah Kunz,

Accountnumber
CU02006293

Extracted Address

Sarah Kunz
Manning Engine Repair Group
212, 92th Street
P.O. Box 10481
N3F 4G7 Manning

Note! Please check that the address fits within the envelope window area and that the address extracted is correct.

Prev 1 2 3 4 5 ... 999 1000 Next

Envelope configuration file

The drop-down list in the Job Information page is automatically populated using the envelope entries in the global_config.json file. The same information is used to draw the window area shown in the PDF preview and it is used to extract the address. The JSON file is found in the config folder located in the workspace folder of the Ad Hoc Mail sample.

```
\workspace\config\global_config.json
```

The envelopes section contains a list of envelope entries. Each describes the name of the envelope type, the window position, the dimensions, the border color shown in the PDF preview and the maximum number of sheets that can go in the envelope. Positions and dimensions are measured in inches.

```
{"envelopes": {  
  "none": {"id": "none", "label": "None"},  
}
```

```

    "C5": {"id": "C5", "label": "C5", "x": "0.79", "y": "2.24", "w":
"3.59", "h": "1.78", "border_color": "LimeGreen", "max_sheets": "3"},
    "DL": {"id": "DL", "label": "DL", "x": "0.75", "y": "2.24", "w":
"3.54", "h": "1.38", "border_color": "DeepSkyBlue", "max_sheets": "2"}
  }
}

```

The startup process of the Workflow configuration copies the contents of this file to the global variable called **config**. We kept the name and JSON structure generic so additional features could be added to the same file.

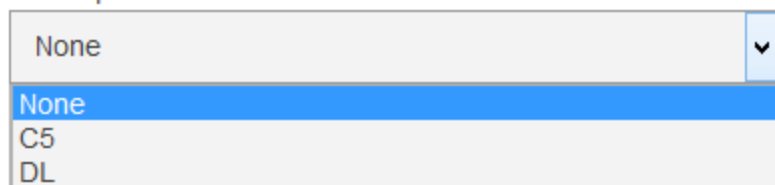
Various processes refer to the **config** variable. They either use the information in the process steps or pass it to the merge engines via a Job Info or in the job data.

Envelope drop-down

The Envelope drop-down shown on the Job Information page is dynamically populated by the envelope list script in the **OLCS-site-tpl** template. It adds the envelope entries found in the config file. This is handled in the following order:

1. The **Job_webpage** process in the Workflow configuration copies the envelope entries from the global **config** variable to **Job Info %7**.
2. The envelope list script in the **OLCS-site-tpl** template file reads the information from this Automation variable. The code iterates over the envelope entries and appends the options to the drop down.

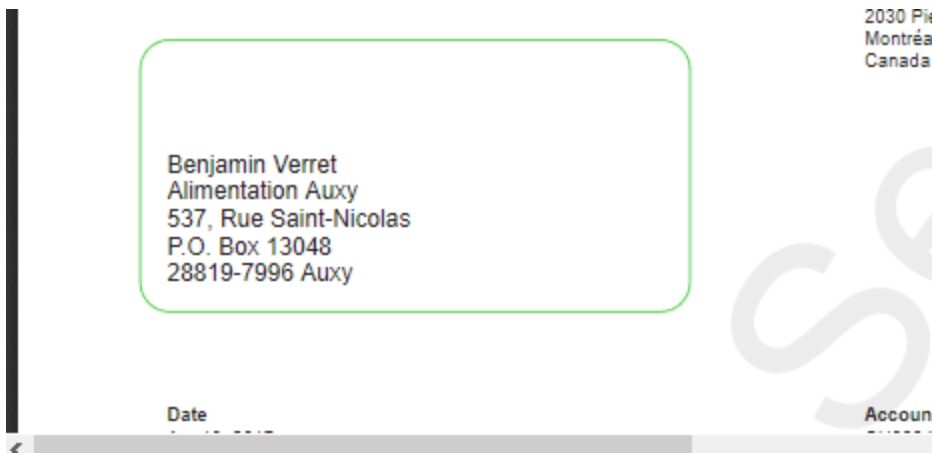
Envelope



The image shows a web-based drop-down menu titled "Envelope". The menu is open, displaying a list of options: "None", "C5", and "DL". The "None" option is currently selected and highlighted with a blue background. The menu has a light gray border and a small downward arrow icon on the right side of the selection box.

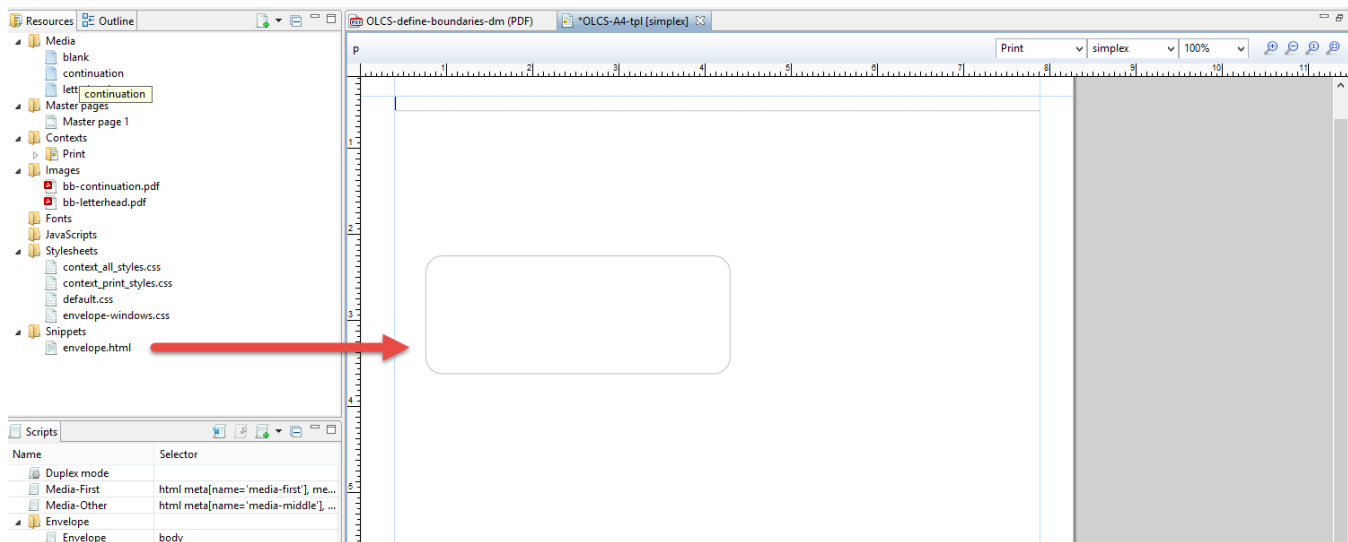
Envelope window area

Selecting an envelope in the Job Information web page generates a PDF preview for the current document. A window area is drawn using the position and dimensions of the selected envelope.



This is handled as follows:

1. The **Job_preview_rest** process in the Workflow configuration reads the parameters of the selected envelope from the global **config** variable and adds this to the job data (which is stored in JSON format). At this stage the job data contains information about the stationery specified by the user, the duplex mode and the selected envelope along with its properties.
2. The job data is written to the **ExtraData** field in the MetaData. (Extracted records and detail tables each have one empty ExtraData field that is added automatically.)
3. The Envelope script in the preview template (e.g. OLCS-A4-tpl) retrieves the job data from the ExtraData field.
4. The script retrieves a generic envelope window element (<div>) from a snippet and adds this to the page. While doing that it sets the position, dimensions and border color.



Address extraction

Selecting an envelope in the Job Information web page invokes a branch in the `Job_preview_rest` process in which the address is extracted. This branch also compares the max number of sheets specified for the envelope with the number of sheets produced for the current document. The address extraction is done based on the input file (e.g. the PS file converted into PDF). It is not using the PDF preview as the stationery and watermark content could be visible in the window area and therefore appear in the extracted address.

The address extraction is handled as follows:

1. The PDF of the input file is read as the Job File.
2. A **Run Script** step retrieves the parameters of the selected envelope from the global config variable. These parameters are used in the Workflow's region functionality. The extracted address is encoded to make sure it is transferred to the browser properly.
3. The address is placed in a local variable.
4. The **Job_preview_rest** process is called via an AJAX request from within the Site template. The return value is a JSON string containing the path to the preview and the extracted address. See the sample below:

```
{ "status": "ok", "preview_url": "/_
iRes/previews/ahjcIxId6E.pdf", "envelope_message":
"", "address": "\u0042\u0065\u006e\u006a\u0061\u006d\u0069\u006e\u002
0\u0056\u0065\u0072\u0072\u0065\u0074\u003c\u0062\u0072\u0020\u002f
```

```
\u003e\u0041\u006c\u0069\u006d\u0065\u006e\u0074\u0061\u0074\u0069\u006f\u006e\u0020\u0041\u0075\u0078\u0079\u003c\u0062\u0072\u0020\u002f\u003e\u0035\u0033\u0037\u002c\u0020\u0020\u0052\u0075\u0065\u0020\u0053\u0061\u0069\u006e\u0074\u002d\u004e\u0069\u0063\u006f\u006c\u0061\u0073\u003c\u0062\u0072\u0020\u002f\u003e\u0050\u002e\u004f\u002e\u0020\u0042\u006f\u0078\u0020\u0031\u0033\u0030\u0034\u0038\u003c\u0062\u0072\u0020\u002f\u003e\u0032\u0038\u0038\u0031\u0039\u002d\u0037\u0039\u0039\u0036\u0020\u0041\u0075\u0078\u0079"} }
```

Adding new envelopes

To add a new envelope:

1. Locate the **global_config.json** file.
2. Open the file in a text editor, for example NotePad++.
3. Duplicate one of the existing entries.
4. Modify the values and save the file.
5. Restart the Workflow server.

Make sure you run the Startup process when debugging your Workflow processes!

Dashboard buttons

The buttons for the dashboard items are located at the top of the Data Table object. Select an entry and click a button on this toolbar to apply the respective action. For more information on Data Tables see <https://datatables.net/>.









Job overview

Print

Preview

Delete

Search:

	Document	Account	UID	ID	Created
	Microsoft Word - mailing.docx 10 pages (1 document)	vandenhe	J2Tflhu0wm	79	2017-05-05
	letter-business bank-1k.pdf 2000 pages (1000 documents)	vandenhe	YhMW97q2En	80	2017-05-05
	Microsoft Word - mailing.docx 10 pages (1 document)	vandenhe	KKXV0JrGsY	66	2017-05-02
	Microsoft Word - mailing.docx 10 pages (1 document)	vandenhe	oHokgVdgjN	68	2017-05-02
	Microsoft Word - mailing.docx 8 pages (1 document)	vandenhe	MxRbUZUYUSK	54	2017-05-01
	Microsoft Word - mailing.docx 10 pages (5 documents)	vandenhe	ErRWEU60CZ	14	2017-04-25
	Microsoft Word - mailing.docx 36 pages (36 documents)	vandenhe	juRibc9ZGy	24	2017-04-25
	Microsoft Word - mailing.docx 20 pages (10 documents)	vandenhe	xfjfsrlGxl	27	2017-04-25

Showing 1 to 8 of 8 entries

Previous **1** Next Approved  Printed

© Copyright Lorem ipsum dolor sit amet

Email notifications

The sample Job page has an email field. When an email address is entered a notification message will be send to the stated email address.

Double Sided
No

Postage
1st Class

Cost Centre
610

Email


Enter your email address and we'll send you a confirmation message!

Extracted Address

Note! Please check that the address fits within the envelope window area and that the address extracted is correct.

Prev 1 Next

Approve



© Copyright Lorem ipsum dolor sit amet

How it works

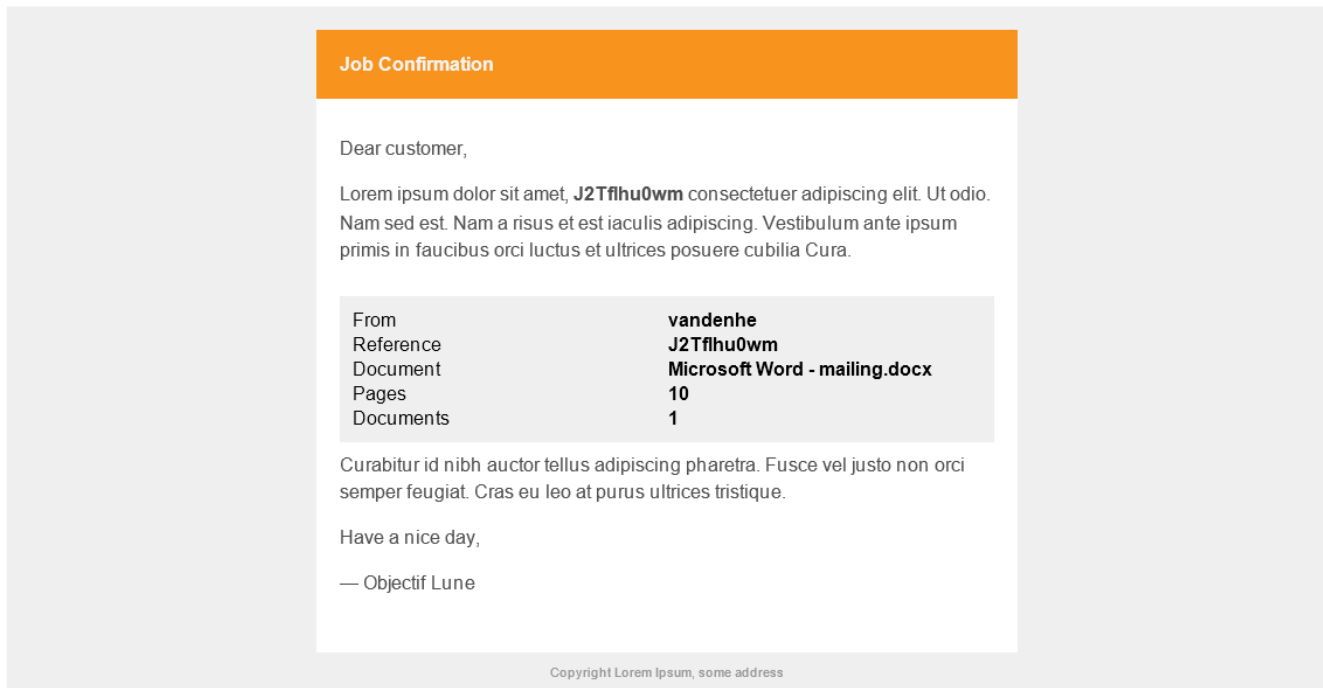
On approval of the job via the Summary_webpage the workflow process copies the Job File to the email folder in the workspace folder of the solution. This folder is monitored by a separate process which sends the confirmation messages.

The **Summary_send_email** process uses the job uid and email address entered to construct an XML file containing the following information:

1. **Uid** The unique identifier of the job is used in the subject and body of the email message.
2. **CSS** The path to the domain specific email CSS stylesheet. This stylesheet is hosted by the domain folder in the workspace directory and is used to apply a domain specific style to the email message. The Dynamic CSS script in the OLCS-email-confirmation-tpl.OL-template adds a <link> element to the <head> of the message referencing this stylesheet file.
3. **Filename** The file name is appended to the subject of the email in the Subject script of the template.
4. **From** We have chosen to script the sender address instead of using the Sender Address stated in the Create Email Content plugin. This provides greater flexibility as this approach allows adding a human readable name by wrapping the actual address between angle brackets. E.g. Peter Parker from OL <pparker@localhost.com>. Alternatively one could state the name of a domain specific account manager or sales rep to further personalize the sender's name (assuming you stored it somewhere in your environment).

5. **To** The email address of the recipient.
6. **JSON** The JSON string containing the parameters set by the user in the Job page.

In case no email address (or an invalid address) was entered the process will not send an email and delete the Job File.



Setting up email notification

Sending the confirmation messages requires to configure the following:

1. Locate the **Summary_send_email** process in the OLCS Workflow configuration.
2. Double-click the **Create Email Content** step.
3. Click the **Email Info** tab.
4. Enter the proper **Mail host** (you can leave the dummy Sender Address as is).
5. Specify the authentication options if needed.

Theming the email

The `OLCS_email_confirmation_tpl.OL_template` implements basic theme switching by dynamically loading the CSS file from the URL provided in the `<css>` data field.

The supplied `email.css` files (located in the domain folders in the workspace folder) change the color of the header bar.

Watermark

The watermark text is added to the JSON string that holds the job parameters (e.g. envelope, stationery etc). This JSON string is passed to the template via the `ExtraData` data field. The `Watermark` script calculates the transformations required to place the watermark text diagonally on the page. The script may look a bit daunting but it does the trick.

Alternatively one could create a simple absolute positioned object on the master page and add static or dynamic text (or logo) and remove the transform code in the script (or the entire script in case of a static watermark).

The color and opacity of the watermark text are specified in the `context_print_styles.css` file of the preview template.

Job information

From	vandenhe
Reference	ALXsJPENlc
Document	Microsoft Word - mailing.docx
Pages	20 pages (1 document(s))

Boundary settings

Single document

Envelope

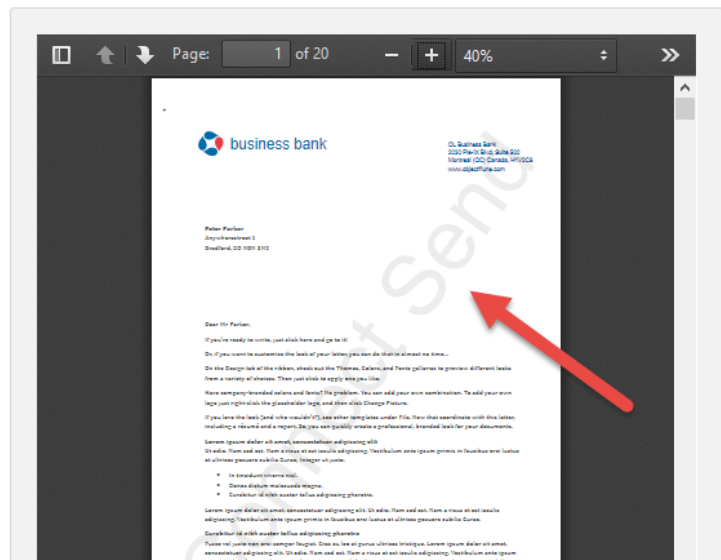
None

Stationery First Page

Letterhead

Stationery Other Pages

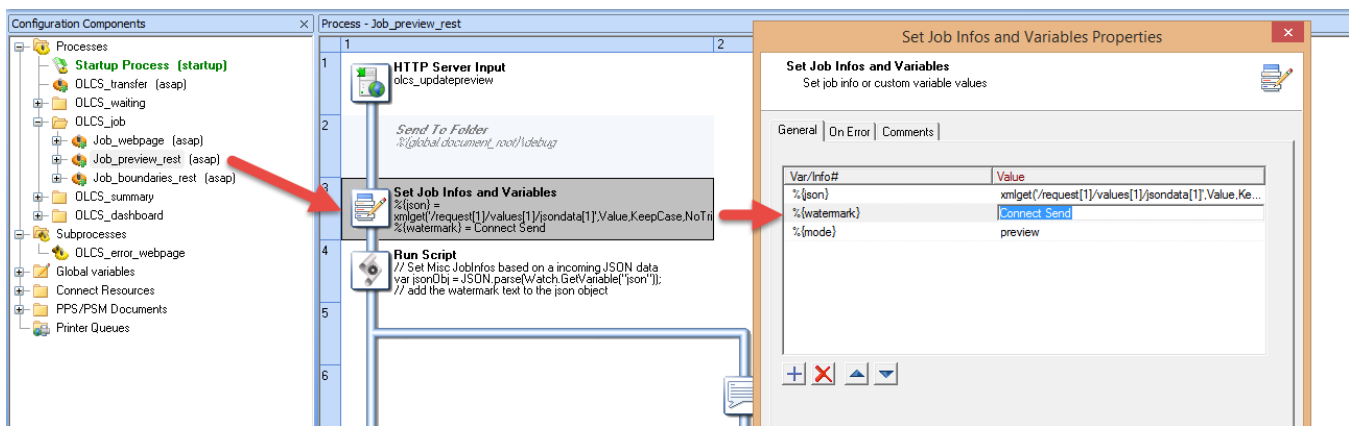
None



Custom watermark text

To change the text of the standard watermark:

1. Open the Workflow configuration.
2. Locate the **Job_preview_rest** process in the OLCS_job folder.
3. Double-click the **Set Job Infos and Variables** step.
4. Change the text for the **%{watermark}** variable.
5. Click OK and send the configuration to the Workflow server engine.



New features included in version 1.1

Below is a list of new features added to the OL Connect Send sample demo .

Duplex modes

A new combo box has been added to the Job Information web page allowing the user to specify if the document should be printed double side or not. This option invokes three modes: simplex (i.e. do not print double sided), duplex without media changes (i.e. the user did not specify any stationery), duplex with media change (the first page is printed on the front of the first media, subsequent pages are print on sheet 2 and on). For more information see: [Duplex modes](#).

Theme switching

Showing a domain specific theme can be implemented in various ways. One could create a web template per domain and add checks in the Workflow processes to dynamically load the template file for that domain. Alternatively one could use the same template file but dynamically load a stylesheet file. The latter is implemented in the latest version of the OLCS-site.tpl file of the Ad Hoc Mail demo. This approach allows for a dynamic change of the color scheme of the pages and makes it possible for example to add a logo image without changing the actual template files. For more information see: [Theme switching](#).

Envelope handling

The Job Information page allows the user to select an envelope for the print job. The drop-down is now automatically populated based on a configuration file. In addition the solution checks if the number of sheets produced will fit in the envelope. In case things do not fit an error message will be shown asking the user to change the job options or select a different envelope type. For more information see [Envelope handling](#).

Dashboard enhancements

Several enhacements were made to the demo dashboard. An option was introduced to show a preview of a job. The options to print and delete jobs are moved to a central location rather than having them per job. This cleans up the user interface. Simply select a job and hit either the Print, Preview or Delete button to apply that option. For a screendump see: [Dashboard](#).

Notification email

The sample Job page has been extended with an email field. When an email address is entered a notification message will be sent to the stated email address. This requires proper set up of the Create Email Content step (e.g. email host name and credentials). For more information see: [Email notification](#).

Watermark

A custom watermark text can be added via a Workflow variable. The respective text is added to the pages in preview mode and automatically scaled to fit the page diagonally. For more information see: [Watermark](#).

Cross browser PDF previews

PDF previews are now shown using the **PDFjs** library assuring it can be displayed across all browsers correctly without installing reader plugins.

Legal Notices and Acknowledgments

Copyright © 2017, Objectif Lune Inc. All rights reserved.

This sample uses the following third party components:

- PDF.js which is distributed under the terms of the Apache Software License Version 2.0. The source code for this can be obtained from the following location: <https://mozilla.github.io/pdf.js/>
- DataTables.js which is distributed under terms of the MIT license. The source code for this can be obtained from the following locations: <https://datatables.net/download/packages>
<https://github.com/DataTables/Plugins/tree/master/integration/foundation>
- PDFObject.js which is distributed under terms of the MIT license. The source code for this can be obtained from the following location: <https://github.com/pipwerks/PDFObject>
- simplePagination.js which is distributed under terms of the MIT license. The source code for this can be obtained from the following location:
<http://flaviusmatis.github.com/simplePagination.js/>