Infor LN Document Output Management User Guide

Publication Information

Release: Infor LN 10.4

Publication date: November 24, 2014

Document code: U9792D US

Contents

About this guide

This guide is written to get you up and running with Infor LN Document Output Management. It covers the basic features and guides you through sending your first document. It also covers the more advanced topics, where you can send documents to your own document management system.

Intended audience

This guide is intended to assist users and system administrators of Infor LN who want to use Document Output Management to send the report output to their preferred destination.

Related documents

You can find the documents in the product documentation section of the Infor Xtreme Support portal, as described in "Contacting Infor" on page 9.

Contacting Infor

If you have questions about Infor products, go to the Infor Xtreme Support portal at www.infor.com/inforxtreme.

If we update this document after the product release, we will post the new version on this Web site. We recommend that you check this Web site periodically for updated documentation.

If you have comments about Infor documentation, contact documentation@infor.com.

Chapter 1 Introduction

1

With the introduction of Document Output Management you can automatically provide each Infor LN Report with the corporate identity of your own company and send it via e-mail to your business partners, store it in Infor Document Management or your own document management system, and/or print it to one or more printers.

By means of printing to the Document Output Management Device the system automatically detects which document must be sent; for example, an Invoice or Order Acknowledgement. With the use of variables on the report, the data values, such as customer name and/or invoice number, will be detected from the document, which can be used in a personalized e-mail.

Besides that, a copy of the document can be sent to the printer, and stored in a document management system. The destination that will be chosen for a document can be configured in the system.

Each document can have attachments, such as Terms & Conditions. Even documents that were sent earlier can be attached to a new document, for example a Reminder can have the original Invoice attached to it.

Using this guide

This guide explains the possibilities of the Infor LN Document Output Management, and how this can improve the way your organization delivers documents. This guide also explains how you can setup the document output for a simple Infor LN Report, but also for more complex situations and how you can send documents to your document management system.

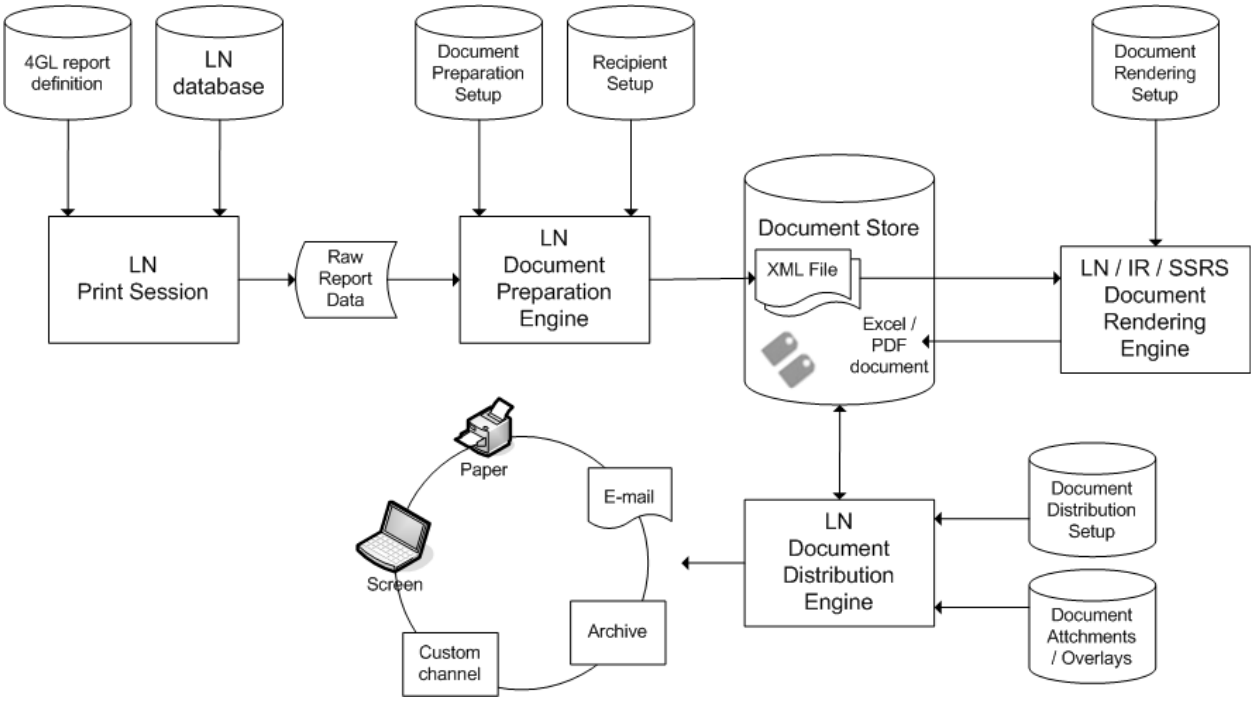
Chapter 2 Overview

2

Introduction

Document Output Management gives you an easy and very flexible solution to send the documents, where your customers define how they want to receive their documents.

This diagram shows the Document Output Management architecture:



This table shows the main components in the diagram:

Component	Description
LN Print Session	The LN Print Session represents any LN Session which collects data from an LN database and produces a report output (raw report data). Document Output Management is designed in such a way, that these LN Sessions don’t need any modification to support Document Output Management.
Document Preparation Engine	The LN Document Preparation Engine generates a batch of documents with routing information. These documents are stored as raw document in the Document Store.
Document Store	The Document Store holds all the documents which were generated by the LN Document Preparation Engine. From the Document Store these documents can be viewed including attachments and overlays. If a document is rendered by the LN Document Rendering Engine, the final document is stored in the Document Store, where it can be distributed by the LN Distribution Engine. It is also possible to repeat a distribution for an already distributed document from the Document Store.
Document Rendering Engine	The LN Document Rendering Engine takes the raw document from the Document Store and processes this to a document in PDF or Excel format. The reporting solutions which can be used by the LN Document Rendering Engine are: Infor Reporting, Microsoft SSRS Reporting and Infor LN’s own PDF / Excel creation.
Document Distribution Engine	The LN Document Distribution Engine takes the document from the Document Store, applies overlays to it, adds additional attachments, and distributes it to the preferred destination: e-mail, printer, Infor Document Management, or a customer defined destination, or a combination of those destinations.

Terms and definitions

Document

A document is a file produced by an LN print session. Each document is stored in the Document Store from where it can be used for further processing.

Batch

A batch is a group of documents produced by one print job.

Document type

A document type describes how a document of a specific category, such as Sales Order Acknowledgement or Invoice, must look when it is distributed to the receiver. A document type contains the email subject and email body text, the overlay page, extra attachments that must be added to the document, and so on.

You can use variables in the parts of the document type. Those variables are replaced by real report values at the moment the document is distributed.

Report rule

A report rule is used to define the report-specific settings, such as how to split up the report in several documents and which report server or report design should be used to render the report. Also the mapping between the report fields and the document type variable is defined in the report rule. You can define multiple Report Rules for one report. Based on the outcome of an expression, one of the report rules is selected. If no expression is valid, the default report rule for that specific report is selected. A report rule links a report to a specific document type.

Receiver type

A receiver is a person or organization who receives the document. Several types are supported:

- Business Partner
- Contact
- Employee

- User
- Custom

Destination type

A destination is the place where to send the documents to. Several types are supported:

- E-mail
- Print
- Custom
- Fax

Document authorization

Users are only allowed to render, distribute, or view documents from the Document Store if they have print authorization for the LN session that produces the document.

Users are not allowed to perform any action on a document if they do not have company authorization for the company in which the document was created.

If a user is not authorized to perform an action, the action is disabled or a message is displayed.

Receiver independent destinations

Usually the receiver type of the document type determines the destination(s) of the document. For example, the document must be sent to the e-mail address that is specified in the business partner data. Besides that, you can also define one or more destinations, which are receiver independent. You want, for example, also to store the documents on your file system. You can specify these destinations on the **Receiver Independent Destinations** tab in the Document Types (ttrpi2510m000) session. Here you can specify the destination type (Email, Print, or Custom) and the destination of the documents.

Rendering and distributing documents

Once available in the Document Store, raw documents can be rendered to a document in PDF or Excel format and distributed to the destinations. Rendering and distributing of documents is handled by the Document Rendering and Distribution (ttrpi3210m000) session.

This session keeps track of the status of each document and the document sequence. According to the status of a document, the document is rendered to a PDF or Excel document, or merged with overlays and attachments and distributed to one or more destinations. Rendering and distribution of documents can be spread over multiple servers (bshells). This session also ensures that a main document is printed together with its subdocuments, and no other documents are printed in between to the same printer.

This session can run in two different modes:

- Run as daemon
- Run for a selection of batches or documents

This session typically runs as daemon in an LN job, started by a system administrator, where it continuously (after a configurable number of seconds of waiting time) searches for new documents to be rendered or distributed.

Viewing documents

Once available in the Document Store, you can view the documents. You can do this by selecting a document and choosing the action ‘View Document’. This will view the document including overlays. Besides this, there are two options to preview documents.

Previewing documents

Usually, documents are rendered and distributed when choosing the Document Output Management device. Afterwards, you can view the distributed document. To preview the document, before it is distributed to the receiver, set the ‘Use During Preview’ field to ‘Yes’ in the Document Type Details session.

When printing the document to the Preview device, a ‘Preview’ batch will be created. When the preview looks fine, the preview batch can be converted to a real batch to be distributed in the normal way. Choose action ‘Convert Preview Batch to Distributable Batch’ in session Batches (ttrpi3500m000).

Easy overlaying

You can preview a document with overlays without setting up the distribution for the document type. This is called easy overlaying and enables you to preview your document with your house style in an easy way.

You must create a document type that has the **Split and Distribute Documents** field set to **No**. When you print the document to the Preview device, the document is stored in the Document Store, but is not distributed to any destination. Once the document is available in the Document Store, you

can preview it including overlays and, if desired, send it to a destination using the “Instant Distribution” functionality. See the next paragraph.

Instant distribution

You can send a document directly to a customer or printer without the intervention of the Rendering and Distribution Engine. This is called instant distribution. To do this, select a document and select the ‘View Document’ action. The Document Viewer starts. Right-click the document and select one of these actions:

- ‘Send to other Printer Destination’. Specify a device and click **Send**.
- ‘Send to other Email Destination’. Specify a valid e-mail address, modify the e-mail body or select a predefined e-mail body, fill the document variables, and click **Send**.
- ‘Send to other FAX Destination’. Specify a valid Fax number and click **Send**.
- ‘Send to other Custom Destination’. Specify Type and address and click **Send**.

Restarting failed documents

Sometimes rendering of a document fails because of an outage of the report server or the distribution of a document fails because of a connection failure to the mail server. In those cases the indicator “Failures Present” will be set to true for that document, and further actions will be blocked.

When the problem has been solved, the status of the documents can be reset and the documents can be rendered and/or distributed again. You can do this by selecting the document and choosing the action ‘Release Failures for Retry’. After this action the document will be selected automatically by the rendering and distribution process. Resetting the status for a range of batches or documents can be done by session Release Failures for Retry (ttrpi3215m000). For more information, see the session help of this session.

Repeating and canceling distribution

Once a document is distributed, this distribution action can be repeated by going to the details of a document, selecting a distribution or destination and choosing the action ‘Repeat Distributions’.

If a document is not yet distributed, its distribution can be cancelled. And if a distribution was cancelled, the cancelation can be undone.

Cleanup document store

Use the Cleanup Batches (ttrpi3200m000) session to remove redundant batches and documents from the document store.

Chapter 3 Getting Started

3

Introduction

This chapter is designed to get you up and running with Document Output Management. It covers some of the basic features and guides you through sending your first Document Output Management document. Once you have worked through this chapter, you can learn about more advanced features using the online Help, or you can continue with the next chapter.

As an example the Request for Quotation report will be send to an email recipient.

Setting up e-mail settings

Before you can use e-mail facilities from Infor LN, you must complete these steps:

1. Start the Service Providers (ttcmf0110m000) session.
2. Check whether the “SMTP” provider exists.
3. If not, create a service provider with these properties:
   - Provider: SMTP
   - Description: SMTP
   - 4GL Connector: ttcmfsmtp
4. On the Specific menu, select Provider Parameters (ttcmf0120m000). Four parameters are generated. Ask your system administrator how to fill in these company-specific parameters.
5. Start the Services (ttcmf0130m000) session.
6. If not available, create a service with these properties:
   - Service Name: SMTP
   - Description: SMTP Service
   - Provider: SMTP
   - Enabled: Yes
   - Logging: Yes
   - Message Storage Path: ${BSE}/tmp/outbox

- File Type: Plain Text ASCII

7 Start the Address Types by Service (ttcmf0140m000) session.

8 If not available, create an Address Type “SMTP” with these properties:

- UI Required: No
- Resolve Capability: Yes
- Service Name: SMTP
- Paper: A4

9 Specify a valid e-mail address (Email Type: SMTP) for the user(s) in the User Data (ttaad2500m000) session.

10 Select the eMessage Connection option in the Parameters (ttrpi2100s000) session.

Setting up fax settings

Before you can use fax facilities from Infor LN, complete these steps:

1. Start the Parameters (ttrpi2100s000) session.
2. Select the Fax Solution options. Select one of these options:
   - Email Faxing: Faxes are sent as ‘normal’ e-mail as defined above.
     Specify a correct e-mail address used for faxing. This address may contain variable values, depending on the used fax solution provider. See Appendix C.
   - Custom Library: You can use an own implementation of a fax solution.

Setting up a simple document type

First step is to create a document type. A document type is used to store the email subject and email body text, the overlay page and extra attachments which should be added to the document. As an example a document type will be created for the Request for Quotation report. Two keywords are used to be able to identify the documents in the Document Store.

To create a document type for the Request for Quotation report:

1 Start the Document Types (ttrpi2510m000) session.

2 Create new Document Type with these properties:

- Document Type: RFQ
- Description: Request for Quotation
- Split and Distribute Documents: Yes
- Receiver Type: User

Note: This distributes the documents to the current user running the Request for Quotation report. This is of course not the Receiver Type you will use in the final setup, but is used now to see the first results of your document type setup.

- Keyword 1, Description: RFQ
- Keyword 1, Zoom Session: tdsls4100m000
- Keyword 1, Zoom Return Field: tdsls400.orno
- Keyword 2, Description: BP
- Keyword 2, Zoom Session: tccom4500m000
- Keyword 2, Zoom Return Field: tccom100.bpid

Save this Document Type record

3 On the **Parts** tab, create a new Part, which will be used as email subject, with these properties:

- Description: Subject
- Destination Type: Email
- Part Type: Email Subject
- Simple Content: Request for Quotation #rfq#

Note: Variables are placed between hashes (#) and will be replaced at runtime

Save this record

4 On the **Parts** tab, create a new Part, which will be used as email body, with these properties:

- Description: Body
- Destination Type: Email
- Part Type: Email Body

Save this record

5 On your local system create a text file with the following content:

```
Dear partner,



Please find attached our Request for Quotation #rfq#.

Our standard Terms & Conditions document is attached as well.



When returning please state #bp#/#rfq#.



With kind regards,
```

#purchaseoffice#

6 Choose action ‘Upload File’ from the just created Document Type Part to upload this text file and add this to the Document Type Part.

Note:

- The **Variables** Tab contains the three variables, which were used in the text file. These variables will be used later on in this chapter, where they will be mapped to report input fields.
- You can create an html file instead of a text file. This html file can also contain the same variables and possibly some pictures and logos.

7 On the **Parts** tab, create a Part, which will be used as filename of the attached document, with these properties:

- Description: Filename
- Destination Type: All
- Part Type: Document Filename
- Simple Content: RFQ_#rfq#.pdf

Save this record

8 On the Mail From Addresses Tab, add an Address, which will be used as From Address of the email, with these properties:

- Mail From Address: noreply@<your.domain>
  Replace <your.domain> with your own company domain, like infor.com
- Mail From Name: Your Company (or any other speaking name for your company)
- Is Default: Yes

Save this record

Setting up a report rule

Next step is to create a report rule to define the report specific settings and map variables and keywords to report input fields.

To create a report rule for the Request for Quotation report:

1 Start the Report Rules (ttrpi2520m000) session.

2 Add a new Report Rule for the Request for Quotation report with these properties:

- Report: tdpur140101002
- Is Default: Yes

- Document Type: RFQ
- Report Server: <empty>
- Render Format: PDF

Save this record

3 Note that on the **Variables** tab, 9 variables have been created with an empty expression.

4 Fill the expressions like the following table, or browse and select the correct report input field.

bp	tdpur105.otbp
keyword1	tdpur105.qono
keyword2	tdpur105.otbp
keyword3	“”
keyword4	“”
purchaseoffice	tdpur012.dsca
receiverfield	logname$
rfq	tdpur105.qono
splitfield	tdpur105.qono & tdpur105.otbp

5 Click **Validate** to validate the settings for this Report Rule and check if the Report Rule is validated and compiled successfully.

Setting up receiver type User

Once you have defined to use receiver type User, complete these steps to define the e-mail addresses and if the document should be (blind) copied to other e-mail addresses as well:

1 Start the User Document Type Settings (ttrpi2551m100) session.

2 Add a new record with these properties:

- Specific User: No
- Specific Document Type: Yes
- Document Type: RFQ

Save this record

3 Double click on the just created record. The User Document Type Settings (ttrpi2551m000) session starts.

4 Add a new record with these properties:

- Destination Type: Email
- From Address Book: Yes

Save this record.

Note: Ensure that your current user's e-mail address has been filled in the address book.

5 Optionally, double click on the just created record. The User Document Type Destinations (ttrpi2552s000) session starts. In this session, you can add extra e-mail addresses to send (blind) copies of the document.

Printing and sending the document

Once you have done all the steps above, the document can be printed and send to the e-mail recipient. Complete these steps:

1 Start the Request for Quotation (tdpur1501m000) session.

2 Select an already existing Request, or create a new one, and select the option to Print Request for Quotations. The Print Request for Quotations (tdpur1401m000) session starts.

3 On the **Device** tab, select the Document Output Management device and click Print. This will print the Request for Quotation and create a document in the Document Store.

4 To distribute the document, start the Document Rendering and Distribution (ttrpi3210m000) session, with these options:

- Run as Daemon: Yes
- Number of Additional Servers: 0
- Query Wait Time: 10

5 Click **Continue**. This will produce a document in PDF format, and distribute this to the e-mail recipient.

Note: Usually this Document Rendering and Distribution session is running in an LN Job, and will continuously search for new documents in the Document Store, which are ready to be rendered and distributed. You can use multiple additional servers to spread the work over multiple bshells.

6 Start the Batches (ttrpi3500m000) session to view the status of the just created batch and its document.

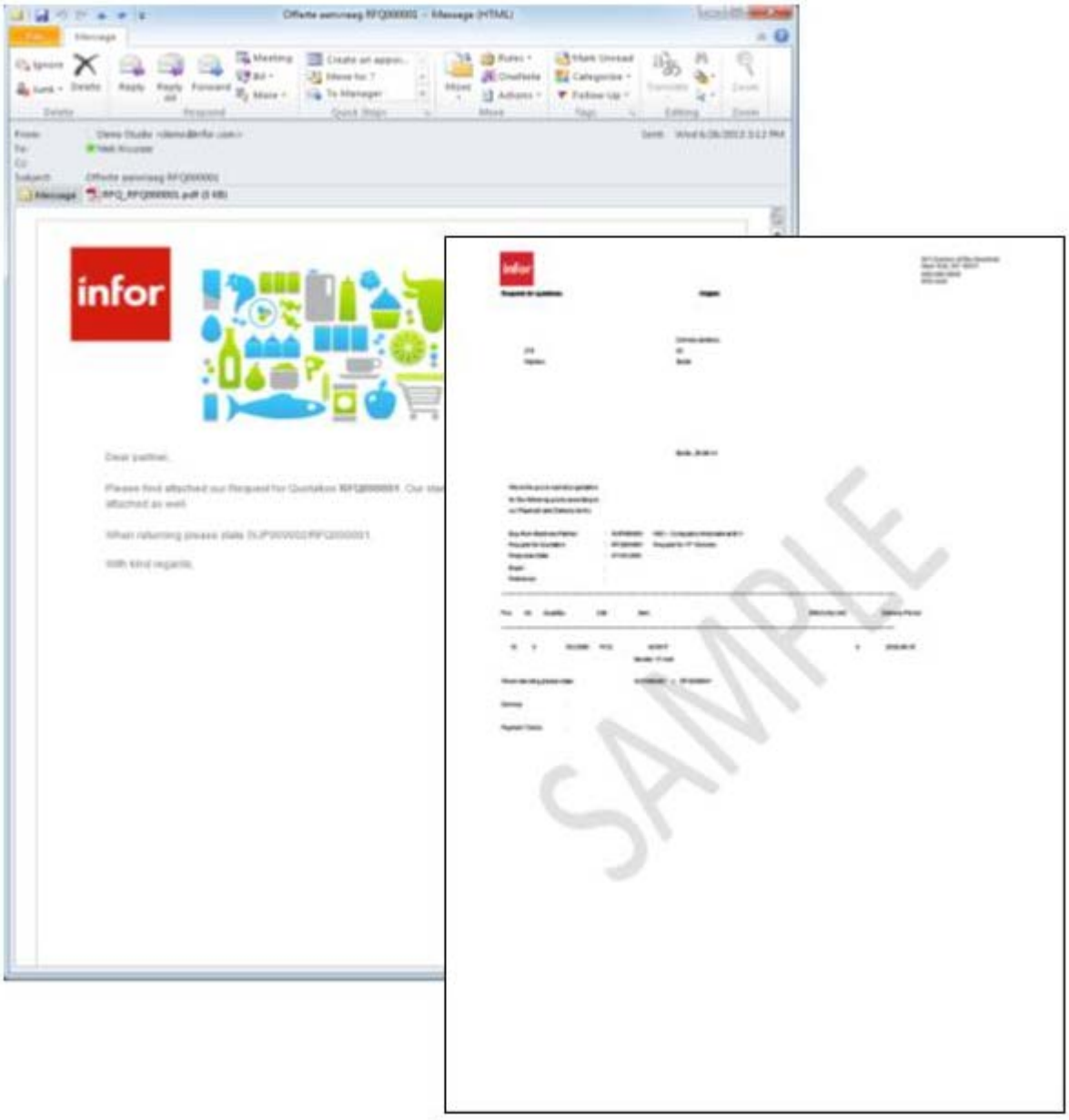
Using overlays

Overlays are used to place additional information on a document. Overlays are PDF files, which will be merged with the document before the document is sent.

For example, suppose you want to make sure that the word COPY appears on the document if the document is sent to a local printer. If the document is sent to a customer, it should contain your corporate identity.

The user can define multiple overlays for one document type. Based on the outcome of an expression, the overlay will be selected. All overlays with a valid expression will be merged to the document.

The following figure shows an example:



To add an overlay page to your document type:

1 Create a one page PDF file containing your company house style, and store it on your local PC.

2 Start the Document Types (ttrpi2510m000) session, and select and open the RFQ document type.

3 On the **Parts** tab, create a new Part with these properties:

- Description: Housestyle
- Destination Type: All
- Part Type: Overlay

Save this record

4 Choose the Upload File Action, select the just created PDF file with your corporate house style and click Open.

5 Print and send the document as above and view the result.

Using attachments

Attachments can be added to the document type. These attachments can either be merged with the document or they can be sent as separate file.

For example, suppose you want to add a price list to your document, which will be merged with the document. And an extra terms and conditions document should to be added to the document as separate file.

The user can define multiple attachments for one document type. Based on the outcome of an expression, the attachment will be selected. All attachments with a valid expression will be merged to the document or added as separate file.

To add an attachment to your document type:

1 Create a PDF file containing terms and conditions, and store it on your local PC.

2 Start the Document Types (ttrpi2510m000) session, and select and open the RFQ document type.

3 On the **Parts** tab, create a new Part with these properties:

- Description: Terms and Conditions
- Destination Type: Email
- Part Type: File to Attach
- Relative Position: After
- Order: 0
- Separate File: No

Save this record

4 Choose the Upload File Action, select the just created PDF file with the terms and conditions and click Open.

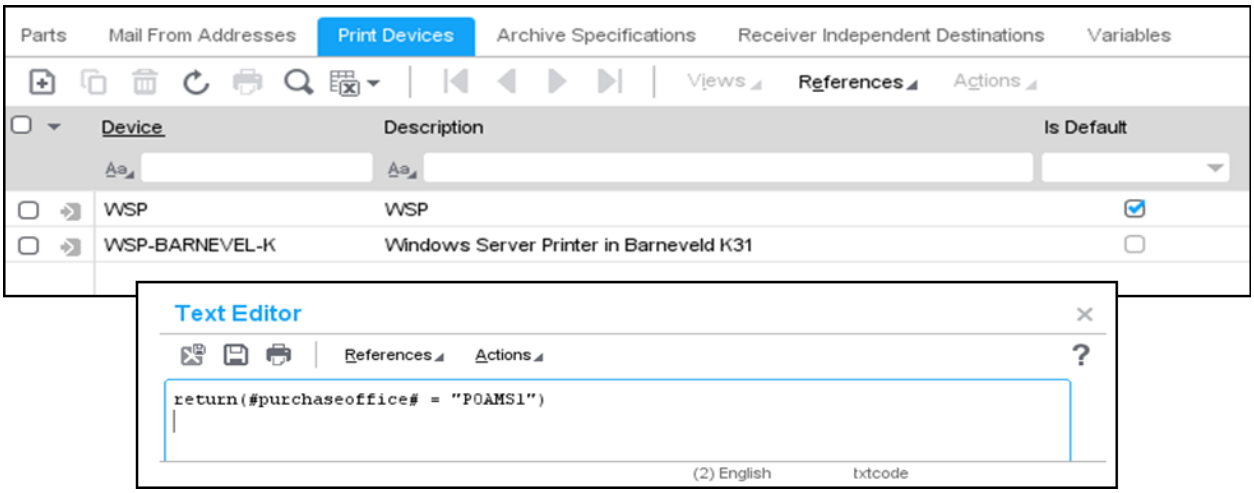
5 Print and send the document as above and view the result.

Using print devices

You can add Print devices to the document type to send the document to a print device.

This is useful, for example, if you want to print the document to a specific device for one purchase office. All other documents are printed on a default device.

The user can specify multiple print devices for one document type. The print device is selected based on the outcome of an expression. If no valid expression applies, the document is printed on the default device.



Note: To test the printing to the correct printer, you must add the destination type Print in the User Document Type Settings (ttrpi2551m100) session.

Setting up receiver type Business Partner

Until now you have used receiver type User and receiver field logname$ (See: To setup a report rule). In that case you were able to send the email to the current LN user. But that is not what you want if you send your request for quotation. The request for quotation should be send to the contact or business partner linked to that request.

To change the receiver type of the document type:

1. Start the Document Types (ttrpi2510m000) session, and select and open the RFQ document type.
2. Change the receiver type to Business Partner, and save the record.
3. Click **Validate** to validate the document type.
4. Start the Report Rules (ttrpi2520m000) session, and select and open the report rule for report tdpur140101002.
5. On the **Variables** tab, change the Expression of the receiverfield variable to: tdpur105.otbp
6. Save the record, and click **Validate** to validate the report rule.

On the other hand the Business Partner data should be extended, to specify the destination (e-mail, print or custom) per business partner for each document type.

To extend the business partner data:

1. Start the Recipient Sets (tccom6140m000) session.

2 Add a new Recipient Set for Recipient Type Business Partner and import all business partners or a selection into the recipient set.

   For more information, see the session help of this session.

3 Start the Document Output Management Details (tccom6170m000) session and add a new record with these properties:

   - Recipient Type: Business Partner
   - Recipient Set: <recipient set>
     (enter the just created recipient set)
   - Document Type: RFQ
   - Destination Type: Mail
   - Mail from Master Data: Yes

   Save the record.

   For more information, see the session help of this session.

After these steps, the document can be send to the business partner, which is linked to the request for quotation. See Print and send a document.

Chapter 4 Advanced Options

4

Introduction

In the previous chapter you have learned some of the basic features of Document Output Management and you have created and sent your first Document Output Management document.

This chapter describes the more advanced features of Document Output Management and several options to enhance the reports and documents. This chapter also describes how you can use different reports or documents by using condition expressions.

Using language dependent distributions

For each document type part, language dependent variants can be defined, so that you send your document with an e-mail body and e-mail subject in the French language to your customers in France while customers in the USA receive their documents and e-mails in the English language.

The language that is chosen for the document type parts is the same as the language in which the Infor LN report is printed.

Preparation

You will use the RFQ document type, created in the previous chapter.

Procedure

You want to send an e-mail in the French language to the customers who are using the French language (system language = 4). All other customers receive the default English e-mail.

To add a language dependent variant to your document part:

1 Start the Document Types (ttrpi2510m000) session, and select and open the RFQ document type.

2 On the **Parts** tab, select the Email Subject part and go to its details.

3 On the **Language Specific Parts** tab, create a new record with these properties:

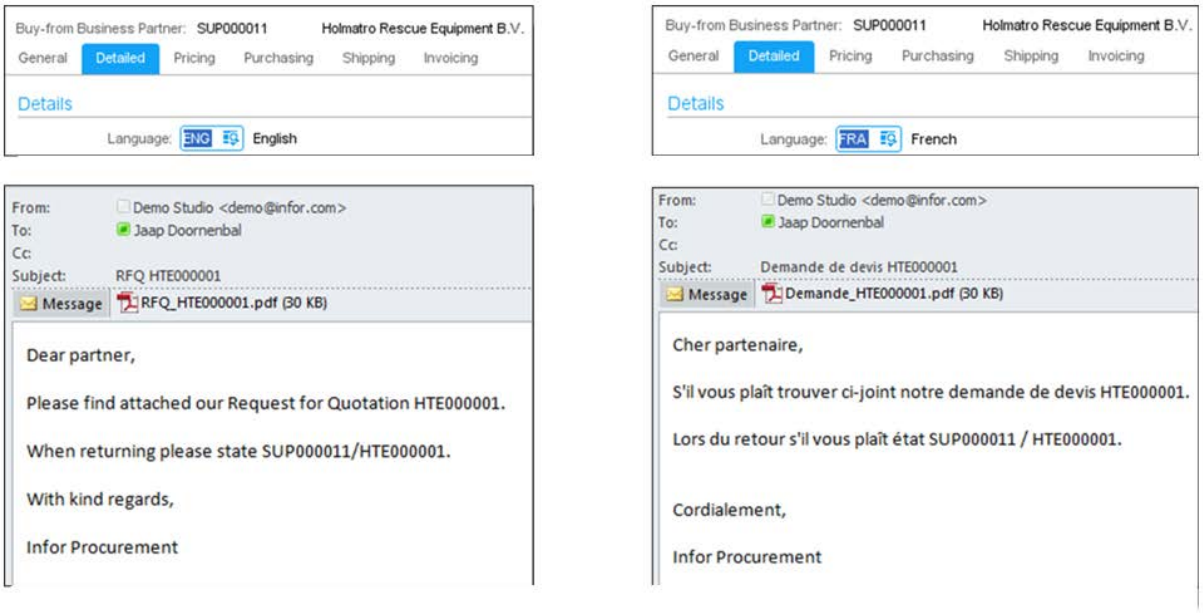
- Language: 4
- Simple Content: Demande de devis #rfq#

Save this record

4 Go back to the Document Type Parts, select the Email Body and go to its details.

5 On the **Language Specific Parts** tab, create a new record for language 4, save the record, and upload a text file or html file containing a French email body text.

This figure shows an example:



Additional information

To get the language in which the document is printed, you can use the dom.document.language variable in expressions linked to the Document Type.

Using configurable designs

You want to use a different report design if the Sales Contract is sent to the US government. Sales Contracts sent to other business partners use the default report design.

Preparation

Create a document type and report rule for the Sales Contract Acknowledgement report and use Infor Reporting to create two different report designs. The default report design is saved as ‘report’. The specific report design is saved as ‘dd250’.

Add a customer defined field, that indicates whether the Business Partner is the US government, to the Business Partner (tccom110) table.

Procedure

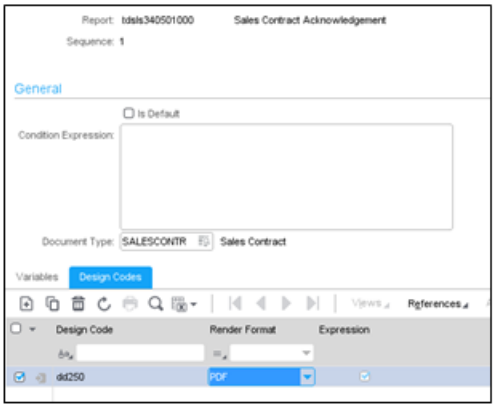
You want to add the two different report designs to the report rule. The specific report design will be used to render the report depending on the outcome of the expression.

Complete these steps:

1 Start the Report Rules (ttrpi2520m000) session. Select the Sales Contract Acknowledgement report (tdsls340501000) and open its details.

2 On the **Design Codes** tab, create a new design code called ‘dd250’ with this expression:

This figure shows an example:

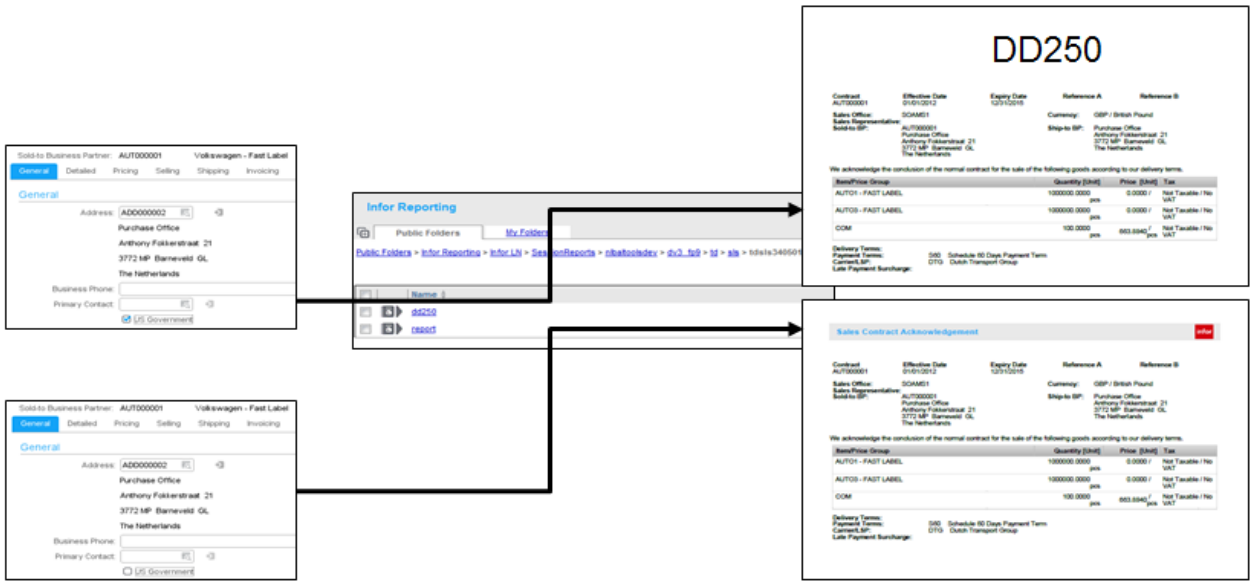


3 Save this record.

4 Click **Validate** to validate the report rule.

5 Print the Sales Contract Acknowledgement report for the US Government to the Document Output Management device and view the result.

This figure shows what the result will be based on the setting of the **US Government** field:



Additional information

Different report designs can be used, based on the outcome of a condition.

In the expression linked to the report design you can use report fields and customer defined fields.

You can also use configurable designs for different countries or languages if not everything can be handled by labels.

Using conditions

Using different rendering options for one report

Suppose you want to use different rendering options for your report. For example, you want to use LN native rendering if RFQ line texts should not be printed on your report. In other cases, you want to render your report using Infor Reporting.

Preparation

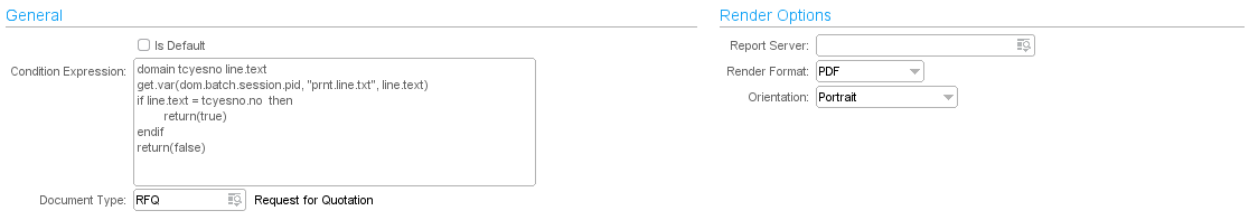
You will use the RFQ document type, created in the previous chapter.

Procedure

You want to define two different report rules for the Request for Quotation report (tdpur140101002). The default report rule was already created in the previous chapter, but must be modified to use Infor Reporting.

Complete these steps:

1. Start the Report Rules (ttrpi2520m000) session. Select report tdpur140101002, and open its details.
2. In the Report Server field select a report server from the available report servers and save this record.
3. Go back to the Report Rules session and add a new rule with these properties:
   - Report: tdpur140101002
   - Is Default: No
   - Document Type: RFQ
   - Report Server: <empty>
   - Render Format: PDF
4. Add this code in the **Condition Expression** field:

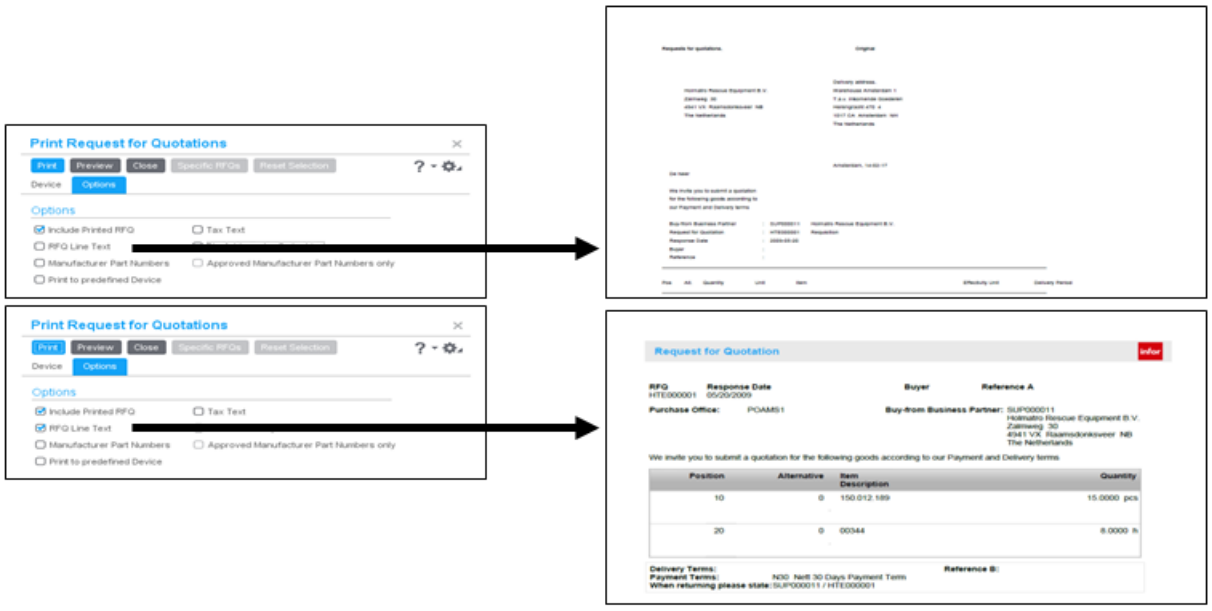


5. Save this record.

   Note: On the **Variables** tab, 9 variables have been created. These variables must get a valid expression. See the previous chapter.

6. Click **Validate** to validate the report rule.
7. Run the Request for Quotations session without Line Texts and with Line Texts. View the results.

   This figure shows what the result will be based on the content of the **prnt.line.text** field on the form:



Additional information

Different report rules can be used, based on the outcome of a condition.

Note:

- In the expression the predefined variable dom.batch.session.pid was used. See appendix B for the complete list of predefined variables.
- In the expression you can also use customer defined fields or form fields of the print session.

Using other overlay for different companies

Suppose you want to use a different overlay for each company in LN. So, if you print an LN report in company 100, you want to use the house style for that company. However, if you print an LN report in company 200, you want to use another house style.

Preparation

You will use the RFQ document type, created in the previous chapter.

Procedure

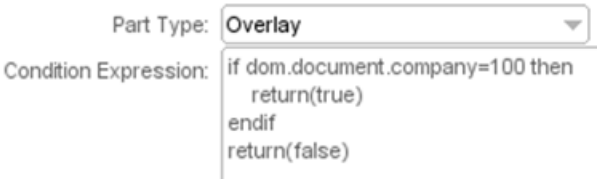
You want to create two different house styles, and add them to the document type. Based on a condition (which LN company is used) one of the two house styles is selected and merged with the document.

Complete these steps:

1 Start the Document Types (ttrpi2510m000) session. Select the RFQ document type and open its details.

2 On the **Parts** tab, open the already existing overlay part, which contains the house style of your company.

3 Add this code in the **Condition Expression** field:

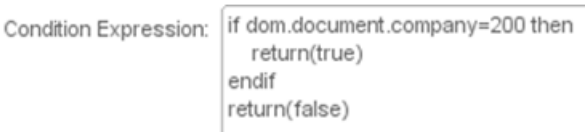


4 Save the record, and return to the Document Type.

5 On the **Parts** tab, add a new record for another overlay part with these properties:

- Description: Housestyle company 200
- Destination Type: All
- Part Type: Overlay
- Overlay Type: All Pages

6 Add this code in the **Condition Expression** field:



7 Save this record.

8 Create a pdf file with the house style for company 200 and upload this file to the new document type part.

9 Click **Validate** to validate the document type.

10 Run the Request for Quotations session in both company 100 and company 200. Print a Request for Quotation to the Document Output Management device and view the results.

Additional information

For each company you can use the house style of that company, without changing the report.

Using other overlay for draft invoice

Suppose you want to use a different overlay for draft invoices, so that the word DRAFT will be added as a watermark to the document.

Preparation

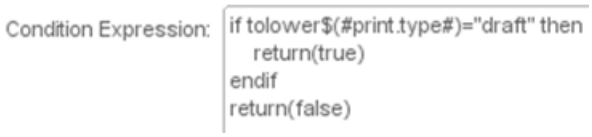
Create a document type and report rule for the Invoice report, or take an already existing document type.

Procedure

You want to create an extra overlay pdf file, which will be added to the Invoice report for draft invoices. Other invoice reports will not get this overlay. Based on a condition, the draft overlay is selected and merged with the document.

Complete these steps:

1. Start the Document Types (ttrpi2510m000) session. Select the Invoice document type and open its details.
2. On the **Parts** tab, add a new record for another overlay part with these properties:
   - Description: Overlay Draft
   - Destination Type: All
   - Part Type: Overlay
   - Overlay Type: All Pages
3. Add this code in the Condition Expression field:

   Save this record.
4. Create a pdf file with a draft watermark and upload this file to the new document type part.
5. Click the Validate button to validate the document type.
6. Note that a new variable is added to the document type, called print.type.
7. Start the Report Rules (ttrpi2520m000) session. Select the report rule for report cisli12001000, which is linked to the Invoice document type. Open the details for this report rule.
8. On the **Variables** tab, click the Refresh Variables button, to refresh the variables from the document type.
9. For the new variable, specify this expression: r.print.type
10. Save this record, and click the Validate button to validate the report rule.

11 Print a draft invoice to the Document Output Management device and view the results.

Using other mail from address for different purchase office

Suppose you want to send requests for quotations from different purchase offices in different locations, where each purchase office uses its own mail-from address. E.g. use mail-from address noreply@infor.com from the purchase office in the USA, and use mail-from address noreply@infor.fr from the purchase office located in France.

Preparation

You will use the RFQ document type, created in the previous chapter.

Procedure

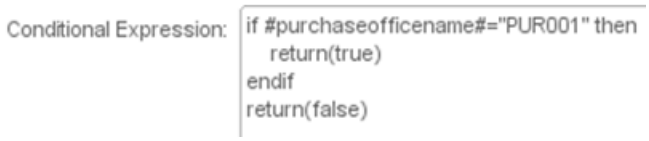
You want to create two different mail-from addresses, and add them to the document type. Based on the name of the purchase office one of the two mail addresses is used as mail-from address in the e-mail message.

Complete these steps:

1 Start the Document Types (ttrpi2510m000) session. Select the RFQ document type and open its details.

2 On the **Mail From Addresses** tab add a new Address and specify a valid email address.

3 Open the details of this new address to add this expression (replace the PUR001 with your own purchase office name):



4 Click the Validate button to validate the document type.

5 This introduced a new variable purchaseofficename, which should be mapped to a report variable. Therefore start the Report Rules (ttrpi2520m000) session and open the details of the report rule of report tdpur140101002.

6 On the **Variables** tab, click the Refresh Variables button, to refresh the variables from the document type.

7 For the new variable, enter this expression: tdpur100.cofc

8 Save this record, and click the Validate button to validate the report rule.

9 Run the LN session Request for Quotations, print two Request for Quotation reports for different purchase offices to the Document Output Management device and view the results.

Additional information

You will notice that different mail-from addresses can be used, based on the outcome of a condition.

Using other printer for different purchase office

Suppose some of your business partners also want to receive a hard-copy of the Request for Quotation document. And suppose your purchase offices in the different locations have its own printer to print the hard-copy.

Preparation

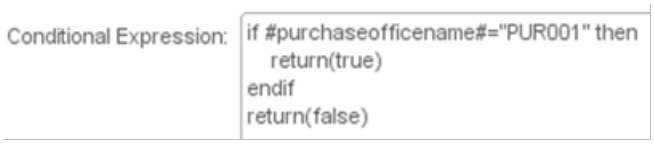
You will use the RFQ document type, created in the previous chapter.

Procedure

You want to create different Print Devices for each purchase office location, and add them to the document type. Based on the name of the purchase office one of the Print Devices is used to create the hard-copy of the Request for Quotation report.

Complete these steps:

1. Start the Document Types (ttrpi2510m000) session. Select the RFQ document type and open its details.
2. On the **Print Devices** tab add a default Device and select one of the available Devices. This default Print Device is used if none of the other Print Devices has a valid expression.
3. Add another Print Device for each of your purchase offices, with this expression (replace the PUR001 with your own purchase office name):



4. Click the Validate button to validate the document type.
5. Run the LN session Request for Quotations, print two Request for Quotation reports for business partners who also wants a hard-copy, from different purchase offices to the Document Output Management device and view the results.

Additional information

You will notice that different printers can be used for different office locations.

Note: You can also use different print devices where each printer has its own preprinted stationary.

Using existing documents

Suppose you want to send a reminder document to your customer with the original invoice attached as additional document. The original invoice was sent earlier, and is still available in the Document Store.

Preparation

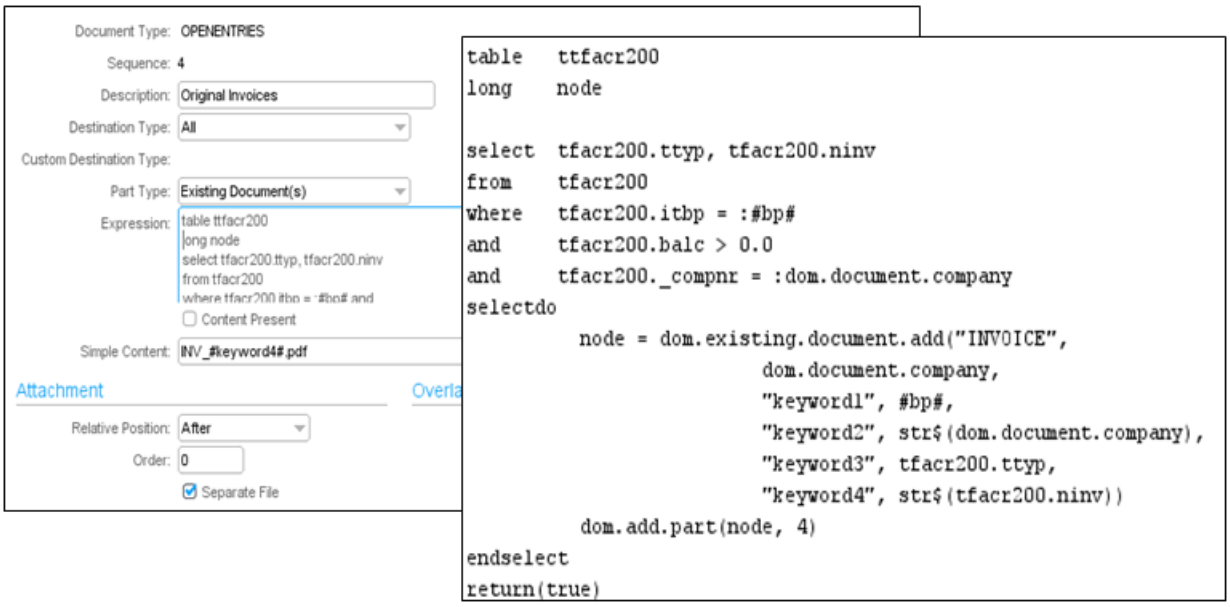
Create a document type and report rule for the Reminder report, or take an already existing document type.

Procedure

You want to add an extra part to the Reminder document type, which will be added as additional document, which contains the original invoice.

Complete these steps:

1. Start the Document Types (ttrpi2510m000) session. Select the Reminder document type and open its details.
2. On the **Parts** tab, create a part with these properties:
   - Description: Original Invoices
   - Destination Type: All
   - Part Type: Existing Document(s)
   - Simple Content: INV_#keyword4#.pdf
   - Relative Position: After
   - Order: 0
   - Separate File: Yes
3. Fill the expression field with this code:



4 Save this record.

5 Click **Validate** to validate the document type.

6 Start the Report Rules (ttrpi2520m000) session to map the new variables to report fields.

7 Click **Validate** to validate the report rule.

8 Print a reminder to the Document Output Management device and view the result.

Additional information

You will notice that an extra document is attached to the e-mail, which contains the original invoice. In the expression of this new part, the existing document is read from the Document Store using the keywords. It is therefore very important to fill the keywords with proper values.

In the expression you are using the function dom.existing.document.add() to retrieve the existing invoice document from the Document Store and add this to your e-mail message. This function has this syntax:

Syntax:

```
function long dom.existing.document.add(
                string i.doctype,
                string i.doc.company,
                [string i.keyword1,
                string i.value1,
                string i.keyword2,
```

string i.value2,

string i.keyword3,

string i.value3,

string i.keyword4,

string i.value4])

Description:

This function retrieves an existing document from the Document Store, including overlays and adds this to the document currently processed.

Arguments:

string i.doctype

The document type of the existing document.

string i.doc.company

The company of the existing document.

string i.keyword1, i.value1, i.keyword2, i.value2, i.keyword3, i.value3, i.keyword4, i.value4

The (optional) string pairs for the keywords and their values, to search the document.

Return value:

This function returns an XML node, which contains the existing document.

You are also using function dom.add.part() to add an overlay to the document. This function has this syntax:

Syntax:

function dom.add.part(

long i.document,

long i.sequence)

Description:

This function adds an extra overlay to the existing document.

Arguments:

long i.document

The xml node which contains the existing document.

long i.sequence

The sequence of the overlay to set

Custom destination types and customer receiver types

Using custom destinations

Besides the currently supported destinations Email and Print, you can also create your own custom destination. This paragraph describes how this can be done and as an example a custom destination type will be created which will store documents on the server file system.

Preparation

Use an existing document type, or create a new one.

Procedure

You want to create a new library, which will store documents on the file system. This library needs to be registered as custom destination type.

Complete these steps:

1 Create a new library tccomdomdestfs with the following content:

```
    domain      tcmcs.str256m     g.path
    domain      tcmcs.str256m     g.folder
    domain      tcmcs.str256m     g.filenames(1) based
    domain      tcmcs.str256m     g.displaynames(1) based
    long              g.nr.files


  function extern long distribution.new(ref string o.error.msg)
  {
    FunctionUsage
    Expl: Initializes distribution.
    Pre:  -
    Post: -
    Input:      -
    Output:     - o.error.msg     - The error message if anything went wrong
    Return: 0 if OK, otherwise the error code
    EndFunctionUsage

    g.path = bse.dir$() & "/dom/"

    g.nr.files = 0

    return(0)
  }


  function extern long distribution.set.document.file(
               const string      i.filename,
```

```
                const string      i.displayname,
                const string      i.mime.type,
                ref string o.error.msg)
    {
      FunctionUsage
      Expl: Sets the document file to be distributed.
      Pre:  -
      Post: -
      Input:      - i.filename      - The file name which contains the document file
            - i.displayname   - The name to be displayed in the distribution
            - i.mime.type     - The MIME type of the document
      Output:     - o.error.msg     - The error message if anything went wrong
      Return: 0 if OK, otherwise the error code
      EndFunctionUsage

      store.file(i.filename, i.displayname)

      return(0)
    }

    function extern long distribution.add.attachment(
                const string      i.filename,
                const string      i.displayname,
                const string      i.mime.type,
                ref string o.error.msg)
    {
      FunctionUsage
      Expl: Adds additional attachment files to the distribution.
      Pre:  -
      Post: -
      Input:      - i.filename      - The file name which contains the attachment
            - i.displayname   - The name to be displayed in the distribution
            - i.mime.type     - The MIME type of the document
      Output:     - o.error.msg     - The error message if anything went wrong
      Return: 0 if OK, otherwise the error code
      EndFunctionUsage

      store.file(i.filename, i.displayname)

      return(0)
    }

    function extern long distribution.set.destination(
                const string      i.address,
                ref string o.error.msg)
    {
      FunctionUsage
      Expl: Sets the destination address.
      Pre:  -
      Post: -
      Input:      - i.address - The address of the destination
      Output:     - o.error.msg     - The error message if anything went wrong
      Return: 0 if OK, otherwise the error code
      EndFunctionUsage

      g.folder = i.address

      return(0)
    }

    function extern long distribution.send(ref string o.error.msg)
    {
      FunctionUsage
      Expl: Stores the received document and attachment on the filesystem
            in the subfolder defined in destination address.
      Pre:  -
      Post: -
      Input:      -
      Output:     - o.error.msg     - The error message if anything went wrong
```

```
    Return: 0 if OK, otherwise the error code
    EndFunctionUsage

    domain      tcmcs.str256m     path
    long  i
    long  ret

    path = strip$(g.path) & strip$(g.folder)
    ret = mkdir(path)

    for i = 1 to g.nr.files
        ret = file.cp(g.filenames(1, i), path & "/" & g.displaynames(1, i))
        if ret < 0 then
            o.error.msg = "Failed to store file " & g.displaynames(1, i)
            return (ret)
        endif
    endfor

    return (ret)
}

function extern long distribution.end(ref string o.error.msg)
{
    FunctionUsage
    Expl: Finalizes a distribution.
    Pre:  -
    Post: -
    Input:      -
    Output:     - o.error.msg - The error message if anything went wrong
    Return: 0 if OK, otherwise the error code
    EndFunctionUsage

    g.nr.files = 0
    free.mem(g.filenames)
    free.mem(g.displaynames)

    return(0)
}

function extern long distribution.cancel(ref string o.error.msg)
{
    FunctionUsage
    Expl: Cancels and removes a distribution.
    Pre:  -
    Post: -
    Input:      -
    Output:     - o.error.msg - The error message if anything went wrong
    Return: 0 if OK, otherwise the error code
    EndFunctionUsage

    return(distribution.end(o.error.msg))
}

function store.file(
            const string      i.filename,
            const string      i.displayname)
{
    domain      tcmcs.str256m     str   fixed

    INC(g.nr.files)
    alloc.mem(g.filenames, len(str), g.nr.files)
    g.filenames(1, g.nr.files) = strip$(i.filename)
    alloc.mem(g.displaynames, len(str), g.nr.files)
    g.displaynames(1, g.nr.files) = strip$(i.displayname)
}
```

2 Start the Custom Destination Types (ttrpi2554m000) session and create a new custom destination type with these properties:

- Custom Destination Type: FILESYSTEM
- Description: Filesystem Destination
- Library: tccomdomdestfs

3 Use this custom destination type to send the document to the file system. How to do this depends on your receiver type:

- If you use receiver type Contact, Business Partner or Employee: start the Document Management Output Details (tccom6170m000) session and add a new record for that receiver type and your document type and destination type ‘Custom’ and custom destination type ‘FILESYSTEM’.
- If you use receiver type User: start the User Document Type Settings (ttrpi2551m100) session and open the details of a specific user or all users and your specific document type. Add a new record with destination type ‘Custom’ and custom destination type ‘FILESYSTEM’.

4 Print to the Document Output Management device and view the result. Note that the documents are stored on the file system in directory <bse_path>/dom.

Additional information

You will notice that it is very easy to create your own destination type and use it for different document types and different recipients. You only have to create a library with these functions, which will be executed by the Document Distribution Engine:

- distribution.new

Syntax:

function extern long distribution.new(ref string o.error.msg)

Description:

Use this function to program the actions to be done before the distribution is created.

Arguments:

o.error.msg: output parameter, which can be filled with an error message if anything went wrong.

Return value: 0 if processing without errors, otherwise the error code

- distribution.set.document.file

Syntax:

```
function extern long distribution.set.document.file(
                        const string i.filename,
                        const string i.displayname,
```

const string i.mime.type,
ref string o.error.msg)

Description:

Use this function to program the actions to be done to add the document to the distribution.

Arguments:

i.filename: input argument with the file name of the document file.

i.displayname: input argument with the display name of the document file.

i.mime.type: input parameter with the mime type of the document file.

o.error.msg: output parameter, which can be filled with an error message if anything went wrong.

Return value: 0 if processing without errors, otherwise the error code

- distribution.add.attachment

Syntax:

function extern long distribution.add.attachment(
const string i.filename,
const string i.displayname,
const string i.mime.type,
ref string o.error.msg)

Description:

Use this function to program the actions to be done to add an attachment to the distribution.

Arguments:

i.filename: input argument with the file name of the attachment file.

i.displayname: input argument with the display name of the attachment file.

i.mime.type: input parameter with the mime type of the attachment file.

o.error.msg: output parameter, which can be filled with an error message if anything went wrong.

Return value: 0 if processing without errors, otherwise the error code

- distribution.set.destination

Syntax:

function extern long distribution.set.destination(
const string i.address,
ref string o.error.msg)

Description:

Use this function to program the actions to be done to add a destination address to the

distribution.

Arguments:

i.address: input argument with the address of the destination.

o.error.msg: output parameter, which can be filled with an error message if anything went wrong.

Return value: 0 if processing without errors, otherwise the error code

- distribution.send

Syntax:

function extern long distribution.send(ref string o.error.msg)

Description:

Use this function to program the actions to be done in case the distribution is sent.

Arguments:

o.error.msg: output parameter, which can be filled with an error message if anything went wrong.

Return value: 0 if processing without errors, otherwise the error code

- distribution.end

Syntax:

function extern long distribution.end(ref string o.error.msg)

Description:

Use this function to program the actions to be done before the distribution is finalized.

Arguments:

o.error.msg: output parameter, which can be filled with an error message if anything went wrong.

Return value: 0 if processing without errors, otherwise the error code

- distribution.cancel

Syntax:

function extern long distribution.cancel(ref string o.error.msg)

Description:

Use this function to program the actions to be done before the distribution is cancelled.

Arguments:

o.error.msg: output parameter, which can be filled with an error message if anything went wrong.

Return value: 0 if processing without errors, otherwise the error code

Note: This technique is very useful if you want to create your own destination or send the documents to your own document management system.

Using custom receiver types

Besides the currently supported receiver types Business Partner, Contact, Employee and User, you can also create your own custom receiver type. This paragraph describes how this can be done and as an example a custom receiver type will be created which is the primary contact of the shipment's carrier.

Preparation

Create a new Document Type SHIPMENT with the necessary parts (email subject, email body, document filename, and mail from address) and create a new Report Rule for the report whinh443511000 with the necessary expressions.

Procedure

You want to create a new receiver type, to be able to send the document to the primary contact of the shipment's carrier.

Complete these steps:

1 Create a new library tcmcsdomcarr with the following content:

```
      table ttccom120   |* Buy-from Business Partners
      table ttccom140   |* Contacts
      table ttcmcs080   |* Carriers/LSP

    #include <bic_dom>

    function extern long dom.get.destinations(
            const string      i.doc.type,
            const string      i.receiver.type,
            const string      i.custom.receiver.type,
            const string      i.receiver.value)
    {
      FunctionUsage
      Expl: Returns the list of destinations
      Pre:  -
      Post: -
      Input:      i.doc.type        - document type
            i.receiver.type   - receiver type
            i.custom.receiver.type  - custom receiver type, if applicable
            i.receiver.value  - receiver value as defined in the report rule
      Output:     -
      Return: destinations as returned by get.dom.document()
      EndFunctionUsage

      long  destination
      domain      tccfrw           carrier
      domain      tcmail           email

      dom.init()

      carrier = i.receiver.value
      email = get.email(carrier)

      if not isspace(email) then
            | send email to the primary contact
            destination = dom.destination.new(DOM_DESTINATION_TYPE_EMAIL)
            dom.destination.set.address(destination, email)
```

```
    endif

    return (dom.get.document())
}

function string get.email(const string      i.carrier)
{
    domain      tcmail      email

    email = ""

    select      tcmcs080.suno
    from  tcmcs080
    where tcmcs080._index1 = { :i.carrier }
    as set with 1 rows
    selectdo
        select      tccom120.ccnt
        from  tccom120
        where tccom120._index1 = { :tcmcs080.suno }
        as set with 1 rows
        selectdo
            select      tccom140.info:email
            from  tccom140
            where tccom140._index1 = { :tccom120.ccnt }
            as set with 1 rows
            selectdo
            endselect
        endselect
    endselect

    return (email)
}
```

2 Start the Custom Receiver Types (ttrpi2554m000) session and create a new Custom Receiver Type with these properties:

- Custom Receiver Type: Carrier
- Description: Carrier
- Library: tcmcsdomcarr

3 Start the Document Types (ttrpi2510m000) session and open the details of Document Type SHIPMENT.

4 Change these properties:

- Receiver Type: Custom
- Custom Receiver Type: Carrier

Save the record and click the Validate button to validate the document type.

5 Start the Report Rules (ttrpi2520m000) session and open the details for report whinh443511000.

6 Change the expression of the receiverfield variable to the following:

```
table  twhinh430
select  whinh430.carr
from    whinh430
where   whinh430._index1 = {:whinh431.shpm}
as set with 1 rows
```

```
selectdo
    return (whinh430.carr)
endselect
return(“”)
```

7 Change the expression of the splitfield variable to: whinh431.shpm

Save the record and click the Validate button to validate the report rule.

8 Start the Print Shipping Discrepancies (whinh4435m000) session, print to the Document Output Management device and view the result.

Additional information

You will notice that it is very easy to create your own receiver. You only have to create a library with this function, which will be executed by the Document Distribution Engine:

- dom.get.destinations

Syntax:

function long dom.get.destinations(
string i.doc.type,
string i.receiver.type,
string i.custom.receiver.type,
string i.receiver.value)

Description:

Use this function to add the destinations to the document.

Arguments:

i.doc.type: input argument which contains the document type

i.receiver.type: input argument which contains the receiver type

i.custom.receiver.type: input argument which contains the name of the custom receiver type, if applicable

i.receiver.value: input argument which contains the receiver value as defined in the report rule

Return value: xml node with the document, including the destinations

In this library some standard defines and functions can be used by including <bic_dom>. See Appendix A for a detailed description of these defines and functions.

Using custom e-mail

In addition to the currently supported CMF as e-mail solution, you can also create your own e-mail solution. This section describes how to do this.

Additional information

To create your own e-mail solution, you must create a library (and specify the library in the Parameters (ttrpi2100s000) session) with these functions, which will be executed by the Document Distribution Engine:

- init.mail.message

Syntax:

function extern long init.mail.message(
ref string o.error.message)

Description:

Use this function to initialize the email message.

Arguments:

o.error.message: output argument which contains the error message, or an empty string

Return value: 0 if processing without errors, otherwise the error code

- set.mail.sender

Syntax:

function extern long set.mail.sender (
const string i.name,
const string i.address,
ref string o.error.message)

Description:

Use this function to add a sender to the email message.

Arguments:

i.name: input argument which contains the name of the sender

i.address: input argument which contains the mail address of the sender

o.error.message: output argument which contains the error message, or an empty string

Return value: 0 if processing without errors, otherwise the error code

- add.mail.recipient

Syntax:

function extern long add.mail.recipient(
const string i.role,
const string i.name,
const string i.address,
ref string o.error.message)

Description:

Use this function to add a mail recipient to the email message. This function can be repeated to add more mail recipients.

Arguments:

i.role: input argument which contains the recipients role. Possible values: to, cc, bcc
i.name: input argument which contains the name of the recipient
i.address: input argument which contains the mail address of the recipient
o.error.message: output argument which contains the error message, or an empty string
Return value: 0 if processing without errors, otherwise the error code

- set.mail.subject

Syntax:
function extern long set.mail.subject(
const string i.subject.
ref string o.error.message)
Description:
Use this function to add a mail subject to the email message.
Arguments:
i.subject: input argument which contains the subject of the mail message
o.error.message: output argument which contains the error message, or an empty string
Return value: 0 if processing without errors, otherwise the error code

- set.mail.body

Syntax:
function extern long set.mail.body(
const string i.body.text.file,
const string i.mimetype,
ref string o.error.message)
Description:
Use this function to add a mail body to the email message.
Arguments:
i.body.text.file: input argument which contains the file name of the body text
i.mimetype: input argument which contains the mime type of the body text
o.error.message: output argument which contains the error message, or an empty string
Return value: 0 if processing without errors, otherwise the error code

- add.mail.attachment

Syntax:
function extern long add.mail.attachment(
const string i.filename,
const string i.displayname,
ref string o.error.message)
Description:
Use this function to add an attachment to the email message.

Arguments:

i.filename: input argument which contains the name of the file.

i.displayname: input argument which contains the name to be displayed in the message

o.error.message: output argument which contains the error message, or an empty string

Return value: 0 if processing without errors, otherwise the error code

- send.mail.message

Syntax:

function extern long send.mail.message(

ref string o.error.message)

Description:

Use this function to send the email message.

Arguments:

o.error.message: output argument which contains the error message, or an empty string

Return value: 0 if processing without errors, otherwise the error code

- cancel.mail.message

Syntax:

function extern long cancel.mail.message(

ref string o.error.message)

Description:

Use this function to cancel the email message.

Arguments:

o.error.message: output argument which contains the error message, or an empty string

Return value: 0 if processing without errors, otherwise the error code

Using custom fax

In addition to the currently supported faxing solution using a fax provider who processes e-mails, you can create your own fax solution. This section describes how to achieve this.

Additional information

To create your own fax solution, you must create a library (and specify the library in the Parameters (ttrpi2100s000) session) with these functions, which will be executed by the Document Distribution Engine:

- init.fax

Syntax:

```
function extern long init.fax(
        ref string o.error.message)
Description:
    Use this function to initialize the fax message.
Arguments:
    o.error.message: output argument which contains the error message, or an empty string
Return value: 0 if processing without errors, otherwise the error code
```

- add.fax.attachment

```
Syntax:
    function extern long add.fax.attachment(
        const string i.filename,
        const string i.displayname,
        ref string o.error.message)
Description:
    Use this function to add an attachment to the fax message.
Arguments:
    i.filename: input argument which contains the name of the file.
    i.displayname: input argument which contains the name to be displayed in the message
    o.error.message: output argument which contains the error message, or an empty string
Return value: 0 if processing without errors, otherwise the error code
```

- set.fax.number

```
Syntax:
    function extern long set.fax.number (
        const string i.fax.number,
        ref string o.error.message)
Description:
    Use this function to set the fax number.
Arguments:
    i.fax.number: The faxnumber to which the fax must be sent.
    o.error.message: output argument which contains the error message, or an empty string
Return value: 0 if processing without errors, otherwise the error code
```

- send.fax

```
Syntax:
    function extern long send.fax(
        ref string o.error.message)
Description:
    Use this function to send the fax.
```

Arguments:

o.error.message: output argument which contains the error message, or an empty string

Return value: 0 if processing without errors, otherwise the error code

- cancel.fax

Syntax:

```
function extern long cancel.fax(
        ref string o.error.message)
```

Description:

Use this function to cancel the fax.

Arguments:

o.error.message: output argument which contains the error message, or an empty string

Return value: 0 if processing without errors, otherwise the error code

Chapter 5 IDM Integration

5

Introduction

Infor Document Management (IDM) is the standard tool within Infor to store documents. With Document Output Management (DOM) it is possible to store documents in IDM or attach an already stored document to the distribution of a document. This chapter explains how documents can be stored in IDM and how they can be reused. The Request for Quotation document from the previous chapter is used as an example.

You can also use your own document solution to store the DOM documents by creating a custom destination, which is described in the previous chapter.

Store documents in IDM

Suppose you want to archive your sent Request for Quotation documents and store them in the IDM archive.

Preparation

You will use the RFQ document type, created in the previous chapter.

Procedure

You want to create a receiver-independent destination and link it to the IDM archive store.

Complete these steps:

1. Start the Parameters (ttrpi2100s000) session and specify the Infor Document Management Settings. Ask your system administrator how to fill in these company specific parameters.
2. Click **Test Connection** to test the connection to the IDM store.
3. Click **Import IDM Metadata** to get and store the available IDM document types and attributes.

4 Start the Document Types (ttrpi2510m000) session. Select the RFQ document type and open its details.

5 On the **Archive Specifications** tab, add a new record with these properties to link an IDM document type to the LN document type:

- SpecificationID: idm
- Description: Archive
- IDM Document Type: MDS_GenericDocument
- LN table: tdpur105

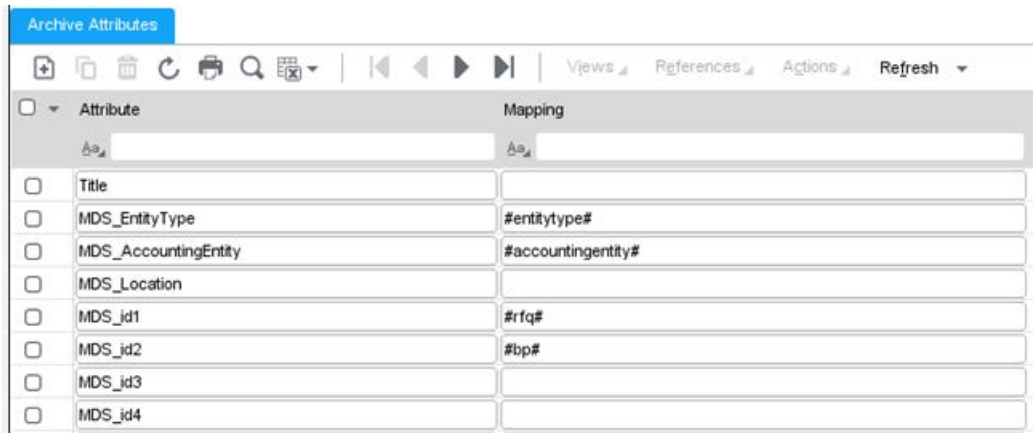
6 On the **Receiver Independent Destinations** tab, add a record with these properties to link an archive specification to a destination:

- Destination Type: Infor Document Management
- Address: idm

7 On the **Attributes** tab, fill the Mapping fields to link the IDM document type attributes to the LN document variables.

Map the MDS_EntityType attribute to the LN table, or use the #entitytype# reserved variable.

Map the MDS_AccountingEntity attribute to the LN company, or use the #accountingentity# reserved variable.



Note: The MDS_id1 – MDS_id2 attributes are reserved for the primary key fields of the LN table. In this example table tdpur105 only has two primary key fields; therefore only id1 and id2 have to be filled.

8 Save this Archive Specification and click **Validate** to validate the document type.

9 Run the Requests for Quotation session. Print a request to the Document Output Management device and verify whether the document is stored in IDM.

Additional information

To use multiple archive specifications, you must create multiple receiver independent destinations on the **Receiver Independent Destinations** tab. The condition expression of the receiver independent destination determines which archive specification is used.

Use existing IDM documents

Suppose you want to send a specific document for the business partner attached to the Request for Quotation document. This specific document is stored in IDM and should be added as additional document to the RFQ document.

Preparation

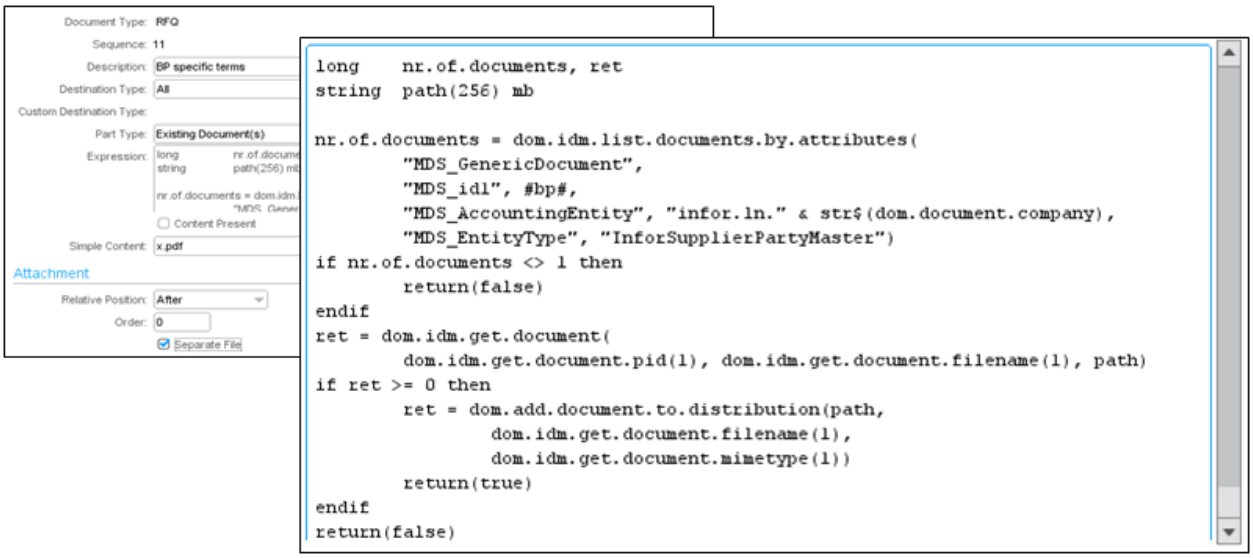
You will use the RFQ document type, created in the previous chapter.

Procedure

You want to add an extra part to the RFQ document type, which will be added as additional document. This additional document is linked to the business partner and stored in IDM.

Complete these steps:

1. Start the Document Types (ttrpi2510m000) session. Select the RFQ document type and open its details.
2. On the **Parts** tab, create a part with these properties:
   - Description: BP specific terms
   - Destination Type: All
   - Part Type: Existing Document(s)
   - Expression:



- Simple Content: x.pdf
- Relative Position: After
- Order: 0
- Separate File: Yes

3 Save this record

4 Click **Validate** to validate the document type.

5 Run the Requests for Quotation session. Print a request to the Document Output Management device and verify the result.

Additional information

You will notice that an extra document is attached to the e-mail, which was linked to the business partner and stored in the IDM archive.

In the expression of this new part, the document is read from the IDM archive using the filename of the document. It is therefore very important to use naming conventions for the file names.

Note: this is a simplified example. In reality there can be multiple documents.

These functions are available when attaching documents from IDM:

- dom.idm.list.documents.by.attributes

Syntax:

function long dom.idm.list.documents.by.attributes (string idm.doctype, …)

Description:

Use this function to retrieve a list of documents from IDM using attributes.

Arguments:

string idm.doctype

The IDM document type

…

A variable number of arguments, containing the name and the value of each attribute

Return value: the number of documents found, otherwise -1

- dom.idm.list.documents.by.xquery

Syntax:

function long dom.idm.list.documents.by.xquery (string i.xquery)

Description:

Use this function to retrieve a list of documents from IDM using XQuery.

Arguments:

string i.xquery

The XQuery string to retrieve a list of documents. See the IDM documentation for the syntax of the XQuery string

Return value: the number of documents found, otherwise -1

- dom.idm.get.document

Syntax:

function long dom.idm.get.document (string i.pid,
string i.filename
ref string o.tempfile)

Description:

Use this function to retrieve one document from IDM and store it as a temporary file.

Arguments:

string i.pid

The internal IDM document ID

string i.filename

The filename (display name) of the IDM document

ref string o.tempfile

(Output) The temporary file (including path), which contains the downloaded IDM document

Return value: 0 if document retrieved, otherwise -1

- dom.idm.get.document.attribute

Syntax:

function string dom.idm.get.document.attribute (long i.entry,

string i.attribute)

Description:

Use this function to retrieve the value of a document attribute from IDM.

Pre:

Call dom.idm.list.documents.by.attributes or dom.idm.list.documents.by.xquery before this method to retrieve the list of documents.

Arguments:

long i.entry

The index of the IDM document

string i.attribute

The name of the attribute which value should be retrieved

Return value: the attribute's value of the document, or empty of an error occurred

- dom.idm.get.document.pid

Syntax:

function string dom.idm.get.document.pid (long i.entry)

Description:

Use this function to retrieve an internal ID of a document from IDM.

Pre:

Call dom.idm.list.documents.by.attributes or dom.idm.list.documents.by.xquery before this method to retrieve the list of documents.

Arguments:

long i.entry

The index of the IDM document

Return value: the ID of the document, or empty of an error occurred

- dom.idm.get.document.filename

Syntax:

function string dom.idm.get.document.filename (long i.entry)

Description:

Use this function to retrieve the filename of a document from IDM.

Pre:

Call dom.idm.list.documents.by.attributes or dom.idm.list.documents.by.xquery before this method to retrieve the list of documents.

Arguments:

long i.entry

The index of the IDM document

Return value: the filename of the document, or empty of an error occurred

- dom.idm.get.document.mimetype

Syntax:

function string dom.idm.get.document.mimetype (long i.entry)

Description:

Use this function to retrieve the mime type of a document from IDM.

Pre:

Call dom.idm.list.documents.by.attributes or dom.idm.list.documents.by.xquery before this method to retrieve the list of documents.

Arguments:

long i.entry

The index of the IDM document

Return value: the mime type of the document, or empty if an error occurred

- dom.add.document.to.distribution

Syntax:

function long dom.add.document.to.distribution (string i.tempfile,
string i.name,
string i.mimetype)

Description:

This function adds an existing document from IDM to the list of distributions.

Arguments:

string i.tempfile

The name of the IDM document to add (including path)

string i.name

The display name of the IDM document to add

string i.mimetype

The mime type of the IDM document to add

Return value: 0 in case of success, otherwise -1

Chapter 6 Troubleshooting

6

Use this troubleshooting information as a resource to help you solve specific problems you may encounter when using Document Output Management.

Logging

When you are troubleshooting, log files can help you.

To get more logging information, you can set these environment variables:

- RPI_SERVER_LOGGING=1

  Logs information from reporting or dom processes in $BSE/log/log.rpi

- DOM_TRACE=<N>

  Logs information from scheduler process in $BSE/log/log.rpi

  - N=1: Errors are logged
  - N=2: Informational messages
  - N=3: Debug level

In case of problems also check the $BSE/tmp/bshell.<pid> file and the $BSE/log/log.bshell file.

Debugging

To detect problems in expressions or custom libraries, you can use Infor LN Application Studio for debugging purposes.

Custom implementations

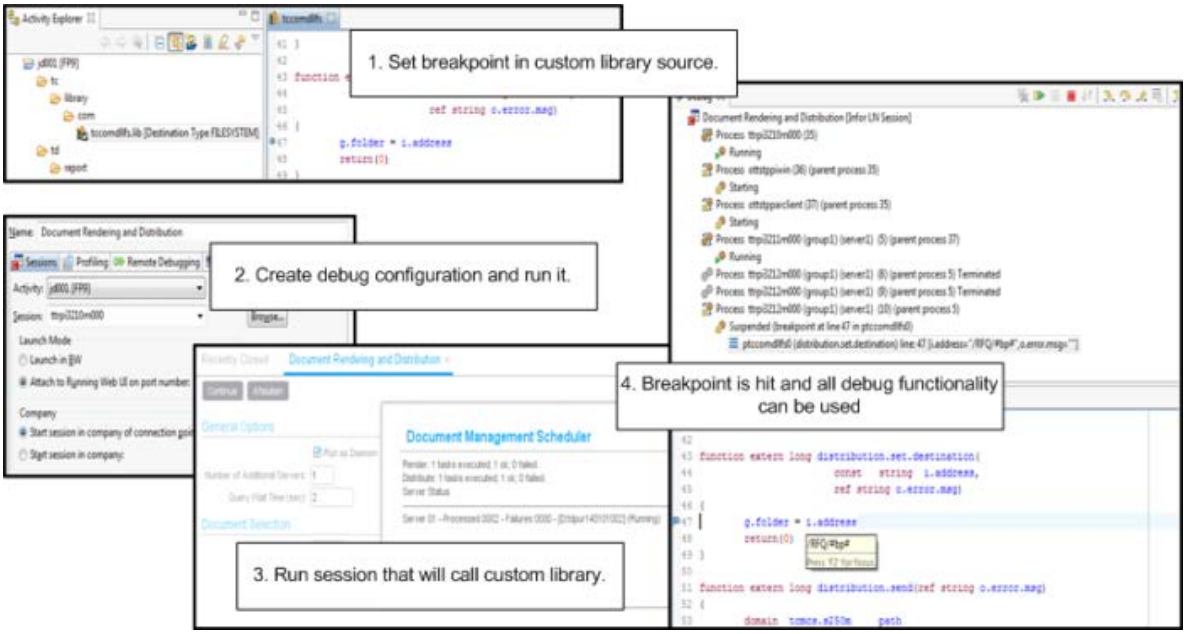
You can debug custom implementations, such as custom destination types, custom e-mail implementations, and custom fax implementations, using Infor LN Application Studio. These custom implementations are libraries that are called by these sessions:

- Document Rendering and Distribution (ttrpi3210m000)
- Documents (ttrpi3510m000) - Select the **View Document** action.

The custom receiver type implementation is called during printing of the original LN report to the Document Output Management device.

Complete these steps:

1. Set breakpoint in library.
2. In case of Web UI, create a launch configuration for one of the sessions mentioned above and launch that session in debug mode from Infor LN Application Studio.
3. In case of LN UI, start the “Debug and Profile 4GL” (ttadv1123m000) session; select your activity and select the debug check box and click **OK**. Start one of the sessions mentioned above from LN UI.



Expressions

You can debug the expressions defined for document types and report rules using Infor LN Application Studio. Actually these expressions are compiled into objects. The generated source code of these document type objects and report rule objects is visible in Infor LN Application Studio.

On the Overview page of the LN Report in Infor LN Application Studio the different Document Output Management Sources, which are applicable for this LN Report, will be displayed. Click on the link related to this source to open the source editor and set a breakpoint.

The document type expressions are called during the preparation phase (printing the LN Report) to determine the printer devices, and during the distribution phase to handle mail from addresses, receiver independent destinations, existing documents, and so on.

Report rule expressions are called during the selection of a device (to select the applicable report rule), and during printing the LN Report to map the correct report values to document variables.

Appendix A Standard Functions

A

bic_dom

The <bic_dom> include contains these defines:

```
#define DOM_DESTINATION_TYPE_EMAIL         "email"
#define DOM_DESTINATION_TYPE_PRINT         "print"
#define DOM_DESTINATION_TYPE_FAX           "fax"
#define DOM_DESTINATION_TYPE_CUSTOM        "custom"
#define DOM_RECEIVER_TYPE_BUSINESS_PARTNER "bus.partner"
#define DOM_RECEIVER_TYPE_CONTACT          "contact"
#define DOM_RECEIVER_TYPE_EMPLOYEE         "employee"
#define DOM_RECEIVER_TYPE_USER             "user"
#define DOM_RECEIVER_TYPE_CUSTOM           "custom"
```

The <bic_dom> include contains these functions:

- dom.init()

  Syntax:

  function extern dom.init()

  Description:

  Initializes the document.

- dom.destination.new()

  Syntax:

  function extern long dom.destination.new (
  const string i.destination.type,
  [const string i.custom.code])

  Description:

  Creates a new destination and adds it to the document created by dom.init(). This can be repeated to add multiple destinations to the document.

  Arguments:

i.destination.type: input variable, which contains the destination type, use one of these values:

DOM_DESTINATION_TYPE_EMAIL

DOM_DESTINATION_TYPE_FAX

DOM_DESTINATION_TYPE_PRINT

DOM_DESTINATION_TYPE_CUSTOM

i.custom.code: input variable, which contains the custom code, if applicable (optional)

Return value:

Returns an xml node containing the newly created destination or 0 (zero) if the destination could not be created.

- dom.destination.set.address()

Syntax:

```
function extern void dom.destination.set.address(
                long i.destination,
                const string i.address)
```

Description:

Sets the address for a given destination.

Arguments:

i.destination: input variable, which contains the destination created by dom.destination.new()

i.address: input variable, which contains the address to set

- dom.destination.set.additional.address()

Syntax:

```
function extern void dom.destination.set.additional.address(
                long i.destination,
                const string i.address,
                boolean i.is.bcc)
```

Description:

Sets the additional address for a given destination. This can be repeated to add multiple additional addresses to the destination.

Arguments:

i.destination: input variable, which contains the destination created by dom.destination.new()

i.address: input variable, which contains the additional address to set

i.is.bcc: input variable (true/false) to set whether the additional address is a bcc (blind carbon copy) or a normal cc (carbon copy)

- dom.get.document()

Syntax:

function extern long dom.get.document()

Description:

Retrieves the document.

Return value:

Returns an xml node which contains the document. In case of an error, the function will return an xml node containing the error message.

- dom.errormessage.set()

Syntax:

function extern void dom.errormessage.set(const string i.error.message)

Description:

Creates an error message for a document. Already added destinations will be removed.

Arguments:

i.error.message: input variable which contains the error message.

Appendix B Using Variables

B

You can use these variables during distribution. You can use them in expressions linked to the Document Type:

- dom.document.language
- dom.document.company
- dom.document.reportcode
- dom.batch.user
- dom.batch.session
- dom.destination.name (the name specified at the Receiver Independent Destination)
- dom.destination.type (the constant names of the ttrpi.dest domain)
- dom.destination.custom
- dom.destination.address
- You can also use the ## expression. The text between the ## is converted to a document type variable and must be mapped to a report variable in the report rule.

You can use these variables in expressions linked to the Report Rule:

- dom.batch.session
- dom.batch.session.pid
- dom.report.language

You can use these functions in expressions linked to the Report Rule:

- get.keyword1()
- get.keyword2()
- get.keyword3()
- get.keyword4()
- get.<report rule variable>()

Appendix C Faxing

C

Depending on the fax solution provider, you must configure the fax settings differently:

- Send an e-mail with attachments.
  - Specify the fax number in the mail address or mail subject.
  - Examples:
    - #FAXNUMBER#@efaxsend.com
    - #FAXNUMBER#-<security code>@informmaxion.faxservice.nl
    - youraccountname@faxservice.nl and mail subject contains <security-code>;#FAXNUMBER#;#subject#
- Implement a web service.
  - XML SOAP web services API