



Infor Enterprise Server Connector for Infor Reporting Development Guide

Copyright © 2016 Infor

Important Notices

The material contained in this publication (including any supplementary information) constitutes and contains confidential and proprietary information of Infor.

By gaining access to the attached, you acknowledge and agree that the material (including any modification, translation or adaptation of the material) and all copyright, trade secrets and all other right, title and interest therein, are the sole property of Infor and that you shall not gain right, title or interest in the material (including any modification, translation or adaptation of the material) by virtue of your review thereof other than the non-exclusive right to use the material solely in connection with and the furtherance of your license and use of software made available to your company from Infor pursuant to a separate agreement, the terms of which separate agreement shall govern your use of this material and all supplemental related materials ("Purpose").

In addition, by accessing the enclosed material, you acknowledge and agree that you are required to maintain such material in strict confidence and that your use of such material is limited to the Purpose described above. Although Infor has taken due care to ensure that the material included in this publication is accurate and complete, Infor cannot warrant that the information contained in this publication is complete, does not contain typographical or other errors, or will meet your specific requirements. As such, Infor does not assume and hereby disclaims all liability, consequential or otherwise, for any loss or damage to any person or entity which is caused by or relates to errors or omissions in this publication (including any supplementary information), whether such errors or omissions result from negligence, accident or any other cause.

Without limitation, U.S. export control laws and other applicable export and import laws govern your use of this material and you will neither export or re-export, directly or indirectly, this material nor any related materials or supplemental information in violation of such laws, or use such materials for any purpose prohibited by such laws.

Trademark Acknowledgements

The word and design marks set forth herein are trademarks and/or registered trademarks of Infor and/or related affiliates and subsidiaries. All rights reserved. All other company, product, trade or service names referenced may be registered trademarks or trademarks of their respective owners.

Publication Information

Release: Infor Enterprise ServerInfor Enterprise Server Connector for Infor Reporting 10.4

Publication Date: February 18, 2016

Document Code: U9751D US

Contents

About this guide.....	5
Intended audience.....	5
Related documents.....	5
Contacting Infor.....	5
Chapter 1: Overview.....	7
Introduction.....	7
Infor ES Connector for Infor Reporting.....	7
Runtime architecture.....	8
Chapter 2: Creating and modifying reports.....	11
Creating a report design - procedure summary.....	11
Creating a report design - procedure details.....	11
Step 1 - Generating an initial report package.....	11
Step 2 - Creating a report.....	12
Step 3 - Modifying the layout of the report.....	13
Step 4 - Adding LN-specific features.....	13
Step 5 – Modifying the prompt page.....	13
Step 6 - Previewing the report.....	14
Step 7 – Importing the report in the LN dictionary.....	14
Resuming work on an existing report.....	14
Chapter 3: LN-specific features.....	15
LN data sources.....	15
LN-specific functions.....	15
Filtering text lines.....	18
tt_filterTextLinewise.....	18
tt_filterTextBlockwise.....	19
Using other label variants.....	20
Using non-report labels.....	21
Barcode images for Infor Reporting.....	21
Examples of barcode images used in Infor Reporting.....	23
Chapter 4: Creating a main report with subreports.....	25
Procedure overview.....	25
Procedure details.....	25
Step 1 - Generate initial report packages for the subreports.....	26

Contents

Step 2 - Design the subreports.....	26
Step 3 - Generate an initial report package for the main report.....	26
Step 4 - Design the main report.....	27
Chapter 5: Viewing and printing reports directly from LN.....	29

About this guide

This document describes how to design Infor LN reports for Infor Reporting.

Intended audience

This document is intended for developers that will create LN reports for Infor Reporting.

Related documents

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

- *Infor Reporting Installation Guide*
- *Infor Reporting Administration Guide*
- *Infor Enterprise Server Connector for Infor Reporting Administration Guide (U9750 US)*

Contacting Infor

If you have questions about Infor products, go to the Infor Xtreme Support portal.

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

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

Introduction

This chapter provides a short introduction of the integration of LN and Infor Reporting.

Infor Reporting provides a complete server-based platform. This platform is designed to support a wide variety of reporting requirements, enabling organizations to deliver information across their entire enterprise.

Infor Reporting runs on a report server. On this server, also the Infor ES Connector for Infor Reporting must be installed.

For details on the installation of Infor Reporting, see the *Infor Reporting Installation Guide*.

For details on the installation of the Infor ES Connector for Infor Reporting, see the *Infor Enterprise Server Connector for Infor Reporting Administration Guide (U9750 US)*.

Infor ES Connector for Infor Reporting

You can use the Infor ES Connector for Infor Reporting to redesign LN session reports for Infor Reporting. Therefore, you can give the reports a modern layout and use features such as images, indicators, and graphs.

You can deploy these reports to a central report server. To use a report design that is stored on the report server, LN users can print the corresponding LN report to an Infor Reporting device.

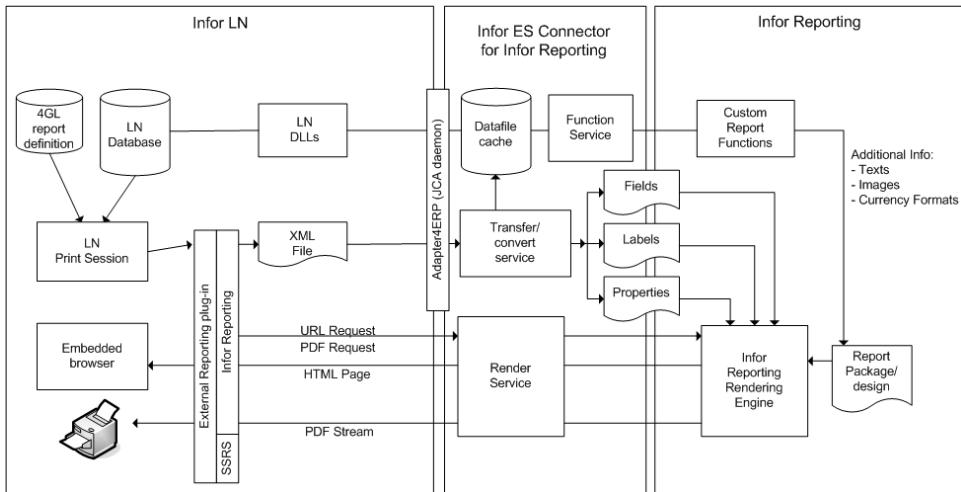
The ES Reporting Connector consists of these components:

- A library, `EEConn4IR.dll`, with LN-specific functions that you can use in your report designs. The library is stored in the `bin` folder of the Infor Reporting installation.
- A web application, `InforErpEnterpriseConnector.war`, in the connector's installation directory.

Runtime architecture

This section describes the components of the integration between LN and Infor Reporting.

This diagram shows the runtime architecture of the Infor Reporting solution for LN:



Note: Currently only push reporting from LN is supported.

The solution for LN contains these parts:

- LN Print Session / 4GL report definition / LN Database

The print session in LN queries the LN database and sends the data to the 4GL report. The 4GL report definition contains the so-called report input fields. These fields are available for the Infor Reporting report.

- External Reporting plug-in / Infor Reporting / SSRS

This is a set of components on the LN host that handles the printing of data by using an Infor Reporting report. The plug-in stores the available Report Servers and handles the flow of data when the user prints an Infor Reporting report. Also, the plug-in provides functions that can be called from the Infor Reporting side to retrieve the (meta)data of the reports.

- XML file

An XML file is created by using the data to be printed on the report. This file contains Report Properties, Labels, and Data Rows. This file is used as the datasource on the Infor Reporting side.

- Embedded Browser / URL Request

When the user views the report directly, a URL Request is built up and activated in the embedded browser of Web UI. Infor Reporting then renders the report and the HTML is sent to the browser.

- PDF stream / Printer

The user can also print the report directly. Therefore, Infor Reporting renders the report in PDF format. This PDF file is sent to the printer.

- Report Package / Design

A report package can contain one or more report designs. The report design is stored on the Infor Reporting side. This design contains the design of the entire report, including the layout, and specifies the datasource that is to be used.

- Infor ES Connector for Infor Reporting

This connector retrieves the XML file from the LN host, through the Adapter4ERP, and splits this file into three separate files that can be processed as datasources by Infor Reporting.

- Infor Reporting Rendering Engine

The rendering engine processes URL requests or Web service requests. The engine runs the query, which must be defined in the report design. For the LN push reports, this query processes the XML files that are created by the Transfer service of the Infor ES Connector for Infor Reporting.

- Adapter4ERP

Adapter4ERP is a common LN component that is used to connect an environment to LN. Adapter4ERP logs on to the server, starts a bshell, and runs BDE- and DLL-calls in LN.

- LN DLLs

During the rendering of the report, it may be necessary to retrieve additional data from the server. You can retrieve additional data through LN-specific functions in the report.

This chapter describes how you can create report designs that are based on existing reports in LN.

Creating a report design - procedure summary

To create and deploy a report design that is based on an LN report:

- 1 Generate an initial report package.
- 2 Create a report.
- 3 Modify the layout of the report.
- 4 Add LN-specific features.
- 5 Modify the prompt page.
- 6 Preview the report.
- 7 Optionally, import the report in the LN dictionary.

See the following sections.

Creating a report design - procedure details

To create a report design that is based on an LN report:

Step 1 - Generating an initial report package

- 1 Log on to LN.
- 2 Start the session for which you will be developing a report.

- 3 Complete the session's form. Ensure that sufficient data is included in the selection ranges. The data that you specify is used only for development and previewing. At runtime, you can specify different selections.
- 4 Print the report. In the Select Device (ttstpsplopen) session, click the **Display** tab and select a device that is configured to generate a file with report metadata and runtime data.

Note:

- In the Infor Workspace LN Plug-in and in classic Web UI, the printer selection may be embedded in the print session. If so, you must select **Advanced Device** and click **Print** to start the Select Device (ttstpsplopen) session.
- Ask your administrator for the name of the appropriate device.
- The device must be of device type "External Reporting Services" and must have this argument: **-DESIGNER -server Infor Reporting Server**
- To create such a device, see the *Infor Enterprise Server Connector for Infor Reporting Administration Guide (U9750 US)*.

When you start printing, various actions are performed automatically, to generate and deploy an initial report package. The Generate and Deploy Report Package window is displayed. This window shows which actions have been completed. If all actions are completed, this message is displayed: "The report preview data file for IR designer has been successfully created and transferred to [Infor Reporting Server] and package has been (re)generated."

- 5 Click **OK** in the message window.

To preview reports, the bshell must stay active. Therefore do not log off from your LN environment.

Step 2 - Creating a report

- 1 Open a browser window, and go to the Infor Reporting portal:
[http://\[servername\]:800/InforReporting/](http://[servername]:800/InforReporting/)
- 2 In the portal, navigate to the report folder on the report server.
This folder is defined in the Report Servers (ttrpi1500m000) session. Ask your administrator for details.
- 3 Click the **Launch** link in the reporting portal's title bar and select **Report Studio**.
- 4 You are now requested to select a package. Select the package for your new report. The package has the code of the 4GL report.
"IBM Cognos Report Studio" is started.
- 5 Click **Create new**. A dialog is displayed.
- 6 Select an Infor template. Complete these steps:
 - a Double-click **Existing....** The "Open" dialog is displayed.
 - b Click **Public Folders** and browse to the Administration folder.
 - c Select one of the templates in the Administration folder, and click **Open**.

- 7 On the **File** menu, select **Save as....** In the "Save As" dialog, browse to your package. Save the report as "report" within the package you selected. The location of the package depends on the folder structure that is defined in the Report Servers (ttrpi1500m000) session. Ask your administrator for details.

Note: the name "report" is required to print the report directly from LN. You can use other names for testing or backup purposes.

You can now start building your report in Report Studio.

Step 3 - Modifying the layout of the report

In the report layout, you can add table columns and various other components. Report Studio has various features to modify the layout of the report. See the IBM Cognos Report Studio documentation.

Step 4 - Adding LN-specific features

You can add various LN-specific features to the report, such as fields, labels, and properties. You can also use LN-specific functions in the report.

See "LN-specific features" on page 15.

Step 5 – Modifying the prompt page

By default, Infor templates have a prompt page, which is normally not used. You can modify this page to store the correct "datafile" value as default, so that you do not have to type it when previewing the report.

Complete these steps:

- 1 Edit the prompt page.
- 2 Delete all parameters, except the first one.
- 3 Click **Add Prompt here**. In the Prompt Wizard, select **Create a new parameter** and specify **datafile**.
- 4 In the Properties of the Text Box Prompt, edit the "Default Selections". Add a value that contains the 4GL report code.
- 5 Save your changes.
- 6 Run the report. The prompt page is displayed. The 4GL report code is already filled in. To view the report page, click **Finish**.

Note: to skip the prompt page, clear the **Prompt for values** check box in the Run with options – report dialog box.

Step 6 - Previewing the report

To generate a preview, run the report.

Step 7 – Importing the report in the LN dictionary

This step is optional. Perform this step if you want the LN export and import mechanisms to distribute the report designs to other environments. See “Import external report designs” in the *Infor Enterprise Server Connector for Infor Reporting Administration Guide (U9750 US)*.

Resuming work on an existing report

To resume work on an existing report:

- 1 Print the report data to the designer device again, just like you did when you started working on the report for the first time.

This step is only required if the report server lost your connection to LN, for example if your bshell was stopped.

Points of attention:

- When printing to the designer device, you are prompted to regenerate the report package. Usually, only the report data must be transferred to the report server. Therefore, click **No** to skip the process of generating the package.

Note: if the 4GL report's input fields have changed, you must regenerate the report package.

- A message is displayed to inform you that the report preview data file is created and transferred. Click **OK** in the message window.
- Do not log off from your LN environment.

- 2 Open a browser window, and go to the Infor Reporting portal:

`http://[servername]:800/InforReporting/`

- 3 Navigate to the report. Points of attention:

- The report is stored in your package.
- The name of the report is “report”.

- 4 To open the report in Report Studio, click the triangular icon beside the report.

Alternatively, click the Launch link in the reporting portal's title bar and start Report Studio. Then click **Open existing** and browse to your report.

This chapter describes how you can use these LN-specific components in your reports:

- LN data sources
- LN-specific functions

LN data sources

On the left side of Report Studio, the "Source" window is displayed. This window contains these data sources:

- Fields
Contains all fields from the LN report.
- Labels
Contains all labels from the LN report.
When you add a label from the Labels dataset, the longest available label of height 1 is used. To use another label variant, see "Using other label variants" on page 20.
- Properties
Contains some general properties, such as "CompanyNumber".

You can use these fields, labels, and properties in report items or field calculations.

LN-specific functions

Various LN-specific functions, which can be used in Report Expressions, are available.

To view the LN-specific functions:

- 1 Create a text field with a Report Expression.
- 2 Go to the **Functions** tab in the left pane of the Report Expression editor.

- 3 In the tree structure in the left pane, expand the Enterprise Server Functions folder.
- 4 To view information in the Tips pane, select a function in the tree.

All functions require the "datafile" as first argument. This value is available as a parameter and must be inserted in the expression as ParamDisplayValue('datafile'). To insert this expression:

- 1 Go to the **Parameters** tab in the left pane of the expression editor.
- 2 Drag "datafile" into the Expression Definition pane.

Alternatively, copy and paste part of the example text from the Tips pane.

This table shows the LN-specific functions:

Function	Syntax and description
tt_formatAmount	<code>tt_formatAmount(datafile, formatCode, currency, amount)</code> Returns a formatted string for the amount, using the given format code and currency. If an invalid code is given, a default-formatted string surrounded by ** is returned (using 2 decimals).
tt_formatCompany	<code>tt_formatCompany(datafile, companyNumber)</code> Returns a string with the formatted company number. The formatting is done according to the LN rules (3 or 4 digits).
tt_getAdditionalFile	<code>tt_getAdditionalFile(datafile, package, module, additionalFile)</code> Returns the URL to an image that is present as an additional file on the LN host. The image is identified by the "package", "module", and "additionalFile" parameters.
tt_getApplicationImage	<code>tt_getApplicationImage(datafile, imageGUID)</code> Returns the URL to an application image that is present on the LN host. The image is identified by a GUID in the data source.
tt_getCompanyLogo	<code>tt_getCompanyLogo(datafile)</code> Returns the URL to the company logo image
tt_getEnumImage	<code>tt_getEnumImage(datafile, domain, value)</code> Returns the URL to an enum image that is present on the LN host. The enum value is identified by the "domain" and "value" parameters.
tt_getLabel	<code>tt_getLabel(datafile, labelCode, length, height)</code> Returns a string with the label variant for the given label code that best fits in the given height and length.
tt_getMessage	<code>tt_getMessage(datafile, messageCode)</code> Returns the message string for the given message code.
ttGetProperty	<code>ttGetProperty(datafile, propertyName)</code> Returns the value for the given property in the Properties dataset.

Function	Syntax and description
tt_getSensitivityImage	<p><code>tt_getSensitivityImage(datafile, height, width)</code></p> <p>Returns the URL to an image, which represents the sensitivity label, with the given height and width in pixels.</p>
tt_getText	<p><code>tt_getText(datafile, textNumber)</code></p> <p>Returns a string with the text that is stored, under the given text number, on the LN host.</p>
tt_getTextFromCache	<p><code>tt_getTextFromCache(datafile, textNumber)</code></p> <p>Returns a string with the text that is stored, under the given text number, on the LN host.</p> <p>Use this function rather than <code>tt_getText</code> if this text number must be printed multiple times on the report.</p>
tt_filterTextLinewise	<p><code>tt_filterTextLinewise(text, includePattern, replacementString)</code></p> <p>Filters the lines of text, replacing lines that match the given <code>includePattern</code> with the given <code>replacementString</code>.</p> <p>Lines that do not match are deleted. The return value contains the filtered text.</p> <p>The <code>replacementString</code> may contain back-references, such as "<code>\1</code>" and "<code>\2</code>", to capturing groups in the <code>includePattern</code>. A <code>replacementString</code> "<code>\0</code>" prints the entire original line without any changes.</p> <p>For examples, see "Filtering text lines".</p>
tt_filterTextBlockwise	<p><code>tt_filterTextBlockwise(text, includeStart, includeEnd, excludeStart, excludeEnd, includeUnmarkedLines)</code></p> <p>Filters the lines of text that are organized in "blocks", and returns the lines that pass the filter.</p> <p>A line that matches the regular expression "includeStart" marks the beginning of an "include block".</p> <p>Similarly, "includeEnd" marks the end of an "include block", and "excludeStart" / "excludeEnd" mark the beginning / end of an "exclude block".</p> <p>All lines within an "include block" are returned by this filter; lines within an "exclude block" are deleted. The start/end marker lines are also deleted.</p> <p>If "includeUnmarkedLines" has value 1, lines outside any block are returned as well; if "includeUnmarkedLines" has value 0, those lines are deleted.</p> <p>For examples, see "Filtering text lines".</p>
tt_getBarcodeImage	<p><code>tt_getBarcodeImage(datafile, value, parameters)</code></p> <p>Returns the URL to an image that represents a barcode for the given value and parameters.</p> <p>See Barcode images for Infor Reporting.</p>

Filtering text lines

This section contains examples that show how you can filter text lines.

A multiline text may contain lines that are marked as "only for internal reports" or "only for external reports". When developing an "external report", you want to filter the text so that the "internal-only" lines are excluded, and the "external-only" lines are included. To accomplish this, you can use one of the "tt_filterText" functions.

These functions are available, for two different types of mark-up used on the text:

- `tt_filterTextLinewise`
- `tt_filterTextBlockwise`

`tt_filterTextLinewise`

This function is used for text on which each line is individually marked as external or internal. For example you can use these special character at the beginning of the lines:

- A "<" character to mark a line as "internal-only"
- A ">" character to mark a line as external-only.

For example, you want to filter this text:

```
Normal line

The empty line above, and this line, are also normal
<This is an internal-line
Normal again
>External line
Final line is normal text.
```

You want the external-only report to have this text:

```
Normal line

The empty line above, and this line, are also normal
Normal again
External line
Final line is normal text.
```

The `tt_filterTextLinewise` function uses these arguments:

- The text to be filtered. This can be a fixed string, but usually this is a `tt_getText` expression.
- A regular expression that matches lines to be included. In the example, the expression must match "normal" and "internal" lines, but should not match "external" lines. This regular expression meets these conditions: `^(>|(?!<))(.*$)`. This expression uses "negative look-ahead" functionality to exclude lines beginning with "<".

- A replacement string for the matching lines. This string may contain "backreferences" to capturing groups in the regular expression. For example, "\2" is substituted by the contents of the second capturing group in the regular expression. "\0" prints the entire matching line.

To use this functionality, specify an expression for a field on your report similar to this code:

```
tt_filterTextLinewise(tt_getText(ParamDisplayValue('datafile'), [your_query].[your_text_field]), '^(>|(?!<))(.*$)', '\2')
```

This code prints these lines:

- Lines that start with ">". The ">" character itself is not included in the output.
- All other lines, except those that start with "<".

The same can be accomplished with this code:

```
tt_filterTextLinewise(tt_getText(ParamDisplayValue('datafile'), [your_query].[your_text_field]), '^>(.*)$|^([<].*)$|^$', '\1\2')
```

In this case the regular expression consists of these alternatives:

- Matching lines that start with ">"
- Lines that start with anything else than "<"
- Empty lines

The replacement string is the concatenation of the first and second capturing groups. At least one of those groups will be empty.

Note: The report must contain a function declaration for the `tt_filterTextLinewise` function.

tt_filterTextBlockwise

This function is used for text consisting of "blocks" that are marked by special begin and end markers. For example, you can use these marker lines:

- "<<<" to mark the beginning and end of an "internal block of text"
- ">>>" to mark the beginning and end of an "external block of text"

For example, you want to filter this text:

```
<<<
This is internal text
consisting of several lines.
<<<
>>>
This is external text
Not to be printed in internal reports
>>>
```

You want the external-only report to have this text:

```
This is external text  
Not to be printed in internal reports
```

The `tt_filterTextBlockwise` function uses these arguments:

- The text to be filtered. This can be a fixed string, but usually this is a `tt_getText` expression.
- A regular expression that matches lines that are used as begin markers for a block of text that must be included.
- A regular expression that matches lines that are used as end markers for a block of text that must be included. In the example, identical begin and end markers are used, but they may be different.
- A regular expression that matches lines that are used as begin markers for a block of text that must be excluded.
- A regular expression that matches lines that are used as end markers for a block of text that must be excluded.
- A flag that indicates whether text that is outside the two blocks must also be included in the output.

This code is an example to use the blockwise filtering:

```
tt_filterTextBlockwise(tt_getText(ParamDisplayValue('datafile'), [your_query].[your_text_field]), '^>>>$', '^>>>$', '^<<<$', '^<<<$', 0)
```

This code prints text inside blocks marked by a start marker ">>>" and an end marker ">>>". All other lines are suppressed.

Note: The report must contain a function declaration for the `tt_filterTextBlockwise` function.

Using other label variants

When you add a label from the Labels dataset, the longest available label variant of height 1 is used.

To use another variant of that label, you must use a different expression in which you specify the available height and length.

To use other label variants:

- 1 Change the “Data Item Value” Source Type into “Report Expression”.
- 2 Delete the existing expression. Remember the field name, because you will require it in the new expression.
- 3 Specify an expression in the form of

```
tt_getLabel(ParamDisplayValue('datafile'), '[labelcode]', [length],  
[height])
```

Where:

- [labelcode] is the fieldname as it occurs in the **Labels** dataset, surrounded by quotes.
- [length] and [height] are numbers that specify the maximum available length and height for the label.

For example, you specify this expression:

```
tt_getLabel(ParamDisplayValue('datafile'), 'tctcmcs002_ccur', 20, 2)
```

Using non-report labels

To use labels that are not part of the original report, such as labels that are not present in the **Labels** dataset, you can use the same expression as described in the “Using other label variants” section:

```
tt_getLabel(ParamDisplayValue('datafile'), '[labelcode]', [length],  
[height])
```

The labelcode to be used here is the complete labelcode as defined in LN, including the package code. For example, specify "tctccom000.nama" to use the "tccom000.nama" label in package tc.

If the label cannot be found, the labelcode is displayed when the report is rendered. In this case, check that you used a valid labelcode and used the correct size for length and height.

Barcode images for Infor Reporting

You can use the `tt_getBarcodeImage` function to include barcode images in reports. You can use this function to call 1D or 2D barcode images.

Syntax

```
tt_getBarcodeImage(datafile, value, parameters)
```

The function returns the URL to an image that represents a barcode for the given value and parameters. The parameters argument is a semicolon-separated string that contains "key=value" pairs. Required keys are "barType", "height", and "width". Depending on the barType, other keys may be required.

Example

The following code returns the URL to a 200x100 image that contains a one-dimensional barcode of type 'BAR39', representing the value '12345':

```
tt_getBarcodeImage(ParamDisplayValue('datafile'), '12345', 'barType=BAR39;height=100;width=200')
```

This table shows the supported Barcode types:

1D Barcode types	2D Barcode types
BAR39	AZTECCODE
BAR39EXT	DATAMATRIX
CODE93	QRCODE
CODE11	PDF417
CODABAR	
CODE93EXT	
CODE128	
EAN13	
EAN8	
EAN128	
INTERLEAVED25	
POSTNET	
UPCA	
UPCE	

This table shows the Barcode Parameters:

Barcode parameter	Barcode Type	Parameter Type	Description
width	all	int	Width of the barcode image.
height	all	int	Height of the barcode image.
fileType	all	string	File type of the barcode image. Supported file types are 'png' and 'jpg'. The default value is 'png'.
codeText	1D	String	Text after encoding. This text contains the appended checksum character, if calculated.
textOnTop	1D	boolean	If true, the text is placed above the barcode.

Barcode parameter	Barcode Type	Parameter Type	Description
checkCharacter	1D	boolean	If true, the checksum character is calculated and appended to the code.
resolution	1D, PDF417	int	Resolution to convert from centimeters to pixels. The value is pixels/centimeter. The default resolution is 96 dpi. If you set the resolution to 1, all units are pixels.
X	1D	double	Size in centimeters of modules (narrow bars or spaces). The resolution is used to convert to pixels.
N	1D	double	Multiplier value for width bars. If the value is 2 (default), wide bars are $2 \times N$ centimeters width.
I	1D	double	Space between two characters (code 39). This is a multiplier of X. The default value is 1.
H	1D	double	Barcode height. This is a multiplier of barcode width. The default value is 0.45.
fit	1D	boolean	If true, the barcode is resized to fit the available area. The default value is false. If both parameters "fit" and "X" are specified, only the height of the barcode is resized.
fontSize	1D	double	Font size of the barcode text. The default value is 11.
fontFamily	1D	string	Font family of the barcode text. The default value is 'Arial'.
dotPixels	DATAMATRIX	int	Size in pixels of the square modules that make up the symbol.
moduleSize	QRCode, Aztec-Code	int	Size (width) of the bars (module). The default value is 4.

Examples of barcode images used in Infor Reporting

To include a Barcode image in an Infor Reporting:

- 1 Open a report from any package in Cognos Report Studio.
- 2 Drag an image from the toolbox and drop it on the desired location in the report design.
- 3 Set the source type of the image to **Report Expression**.
- 4 Click the ellipses button for the **Report Expression**.
- 5 Click the 'functions' icon and expand the Enterprise Server Functions.
- 6 Double-click the **tt_getBarcodeImage** function.
- 7 Specify an appropriate expression.

For example, specify one of these expressions:

- ```
tt_getBarcodeImage (ParamDisplayValue('datafile'),
 '78787879873', 'barType=BAR39;width=400;height=200;fileType=
 png')
```

Specify this expression to include a 400x200 image that contains a one-dimensional barcode of type 'BAR39', representing the value '78787879873':



78787879873

- ```
tt_getBarcodeImage (ParamDisplayValue('datafile'),
    '78787879873', 'barType=QRcode;moduleSize=8;width=200;height=200;fileType=png')
```

Specify this expression to include a 200x200 image that contains a two-dimensional barcode of type 'QRcode', representing the value '78787879873':



Creating a main report with subreports

4

This chapter describes how to create a main report with one or more subreports for an LN session. Creating a main report with subreports is necessary for these sessions:

- Sessions for which one 4GL report is opened and closed multiple times within a single print batch. An example is the Print Contract Quotation Documents (tsctm2400m000) session.
- Sessions for which multiple 4GL reports are printed within a single print batch. For example, the Print Production Order Documents (tisfc0408m000) session can print various reports/documents within a single print batch, such as these:
 - Order Covering Note (tisfc040801000)
 - Routing Sheet (tisfc040802000)
 - Operation Note (tisfc040803000)
 - Material List (tisfc040804000)

Procedure overview

To create a main report with subreports:

- 1 Generate initial report packages for the subreports.
- 2 Design the subreports.
- 3 Generate an initial report package for the main report.
- 4 Design the main report.

Procedure details

Complete these steps:

Step 1 - Generate initial report packages for the subreports

To generate initial report packages for the individual subreports, you must print the corresponding LN reports/documents, one by one, to a designer device.

Repeat these steps for each subreport:

- 1 In LN, start the session for which to create the main report and subreports.
For example, start the Print Production Order Documents (tisfc0408m000) session.
- 2 Complete the session's form and select the report/document for which you want to create a subreport.
For example, in the Print Production Order Documents (tisfc0408m000) session, you only print the Order Covering Note (tisfc040801000) report for one production order.
Print the report to an Infor Reporting designer device. This is a device of type "External Reporting Services." The device must have this argument: -server [Infor Reporting Server]
Various actions are performed automatically and a message is displayed.
A report package is generated for the subreport. For example, tisfc040801000.

Step 2 - Design the subreports

Log on to the Infor Reporting server and modify the layout of the subreports.

In the report layout, you can specify page margins and add table columns, a page header, a footer, a company logo, and various other components.

Report Studio has various features to modify the layout of the report. See the IBM Cognos Report Studio documentation.

Step 3 - Generate an initial report package for the main report

To generate an initial report package for the main report:

- 1 In LN, start the session for which to create the main report and subreports.
For example, start the Print Production Order Documents (tisfc0408m000) session.
- 2 Complete the session's form and select at least two reports for which you want to create a subreport.
For example, in the Print Production Order Documents (tisfc0408m000) session, you print the Order Covering Note (tisfc040801000) and Operation Note (tisfc040803000) reports.
Print the reports to an Infor Reporting designer device. Various actions are performed automatically and a message is displayed.
- 3 Click **OK** in the message window. Do not log off from your LN environment.
This report package is generated: [LN main report name]_main
For example: tisfc040801000_main

Step 4 - Design the main report

When printing, images are generated for the subreports. These images must be displayed in the main report. To achieve this, the main report layout must contain an image control that is linked, through a query item, to the subreport images.

Note: Do not define a page header, footer, page margins, company logo, and so on in the main report. The main report must have a stripped layout, because the subreport images that will be displayed on the report at runtime have their own page headers, footers, and so on.

To design the main report:

1 In Infor Reporting, edit the layout of the main report. The main report is stored in the [LN main report name]_main report package.

2 Use the toolbox to add a list in the report layout.

The list must have one column. The column must have no title and no borders.

3 Use the toolbox to add an image control in the list.

4 Add a query item to the report. Complete these steps:

a In the Cognos Report Studio, open the Query Explorer.

b In the Query Explorer, double-click the “Query1” item.

c Open the Fields node in the Source view.

d Double-click the “imageUrl” item in the Source view.

A query item is created. The query item is displayed in the Query Explorer.

The query item is named “imageURL” and has this expression: [Model].[Field].[imageUrl]. Verify this in the Properties view of the query item.

5 Assign the query item to the image control. Complete these steps:

a Select the image control in the list.

b In the properties list, under **URL Source**, specify this information:

Source Type

Data Item Value

Data Item Value

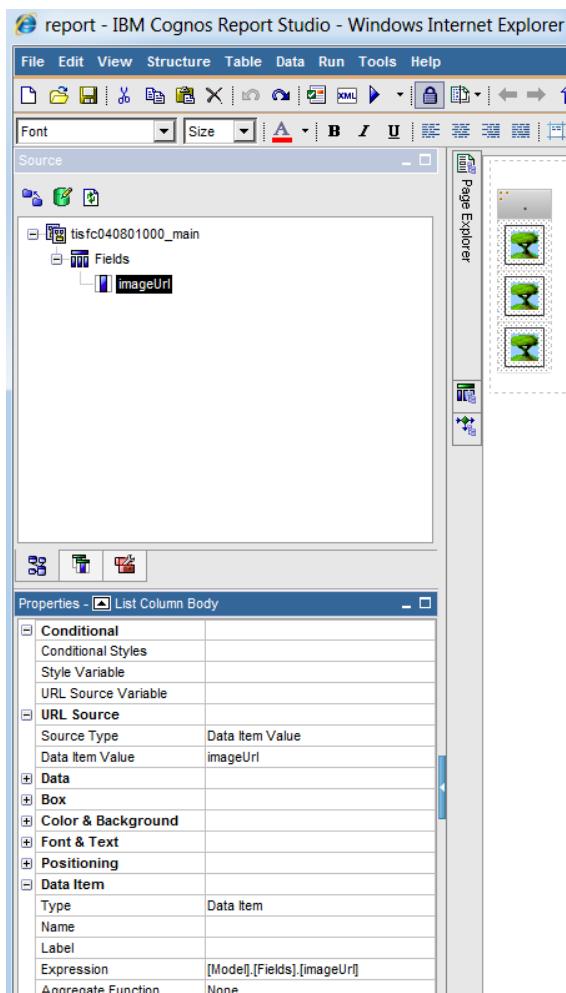
imageUrl

Now the image control is linked, through the query item, to the subreport images.

6 Save the report.

This diagram shows an example:

Creating a main report with subreports



Viewing and printing reports directly from LN

5

To print a deployed report from LN:

- 1 Log on to LN.
- 2 Start the session to which the LN report is linked.
- 3 Complete the session's form. Ensure that sufficient data is included in the selection ranges.
- 4 Print the report. In the Select Device (ttstpsplopen) session, click the **Display** tab and select an appropriate device.

Note:

- Ask your administrator for the names of the appropriate devices.
- The device must be of device type "External Reporting Services," and must have one of these arguments:
 - “-server [server name]”: This device renders the report on the screen.
 - “-server [server name] -printer [printer name]”: This device directly sends the report to the printer without rendering the report on the screen first.
- For details on how to create such a device, see the *Infor Enterprise Server Connector for Infor Reporting Administration Guide (U9750 US)*.

