M3 Business Engine Configurable Output Management Configuration Guide

Version 15.1.4
Published March 20, 2019

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

Publication Information

Release: 15.1.4

Publication date: March 20, 2019

Document Number: M3BECOMCG_15.1.4_UWA_05

Contents

About this guide

1

This guide provides information on how to configure and install M3 Business Engine Configurable Output Management or M3 BE COM.

Prerequisite knowledge

To fully understand the information presented in this guide, you must have knowledge and experience in these areas:

- OpenText StreamServe 5.6.2 or OpenText Exstream 16.4
- XMLIN (Extensible Markup Language)
- StoryTeller

  Note: This is only applicable when using the XML StoryTeller Base project in StreamServe.
- Windows system administrator experience

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

2

The M3 Business Engine Configurable Output Management is paving the way for easier document management and improved extensibility.

For M3 BE COM, the output is an XML file instead of a Streamfile. One advantage of using a configurable XML file is that additional fields can be added to the output without the need for Java coding. Layout templates are provided in OpenText StreamServe StoryTeller instead of PageOut.

M3 Configurable Output Management is only available in M3 BE 15.1.4 MCP2.

Installation package overview

Download the file M3_BE_15.1.4 Configurable Output Management vxx.zip from Infor Download Center or Xtreme Knowledge Base article 1912248. The file includes these zip files:

- M3_BE_15.1.4_XML_StoryTeller_Base_Project_Template_vxx.zip
- M3_BE_15.1.4_IDM_Configuration_vxx.zip

  These two files are only required when using the Russia country version:

- M3_BE_15.1.4_ConfigData_RussiaDocuments_vxx.zip
- M3_BE_15.1.4_XML_StoryTeller_Forms_Project_Template_vxx.zip

All zip files include a README PDF file with a detailed list of content, change log, and limitations.

For a list of included base layouts, see REAME file in M3_BE_15.1.4_XML_StoryTeller_Base_Project_Template_vxx.zip.

For a list of included form layouts, see REAME file in M3_BE_15.1.4_XML_StoryTeller_Forms_Project_Template_vxx.zip.

Note: Streamfile is a proprietary file format consisting of IDs and values, and is currently used as an output method for M3 BE.

M3 BE COM is delivering changes and new document through Xtreme Knowledge Base article 1912248. Use the KB's "Sign Up" feature to monitor updates through mail notifications. The installation package (zip file) is found as attachments on the KB article, accompanied by an "Update Notification" table to list the latest changes.

Periodically, the installation package is posted on Infor Download Center for new customers to pick up.

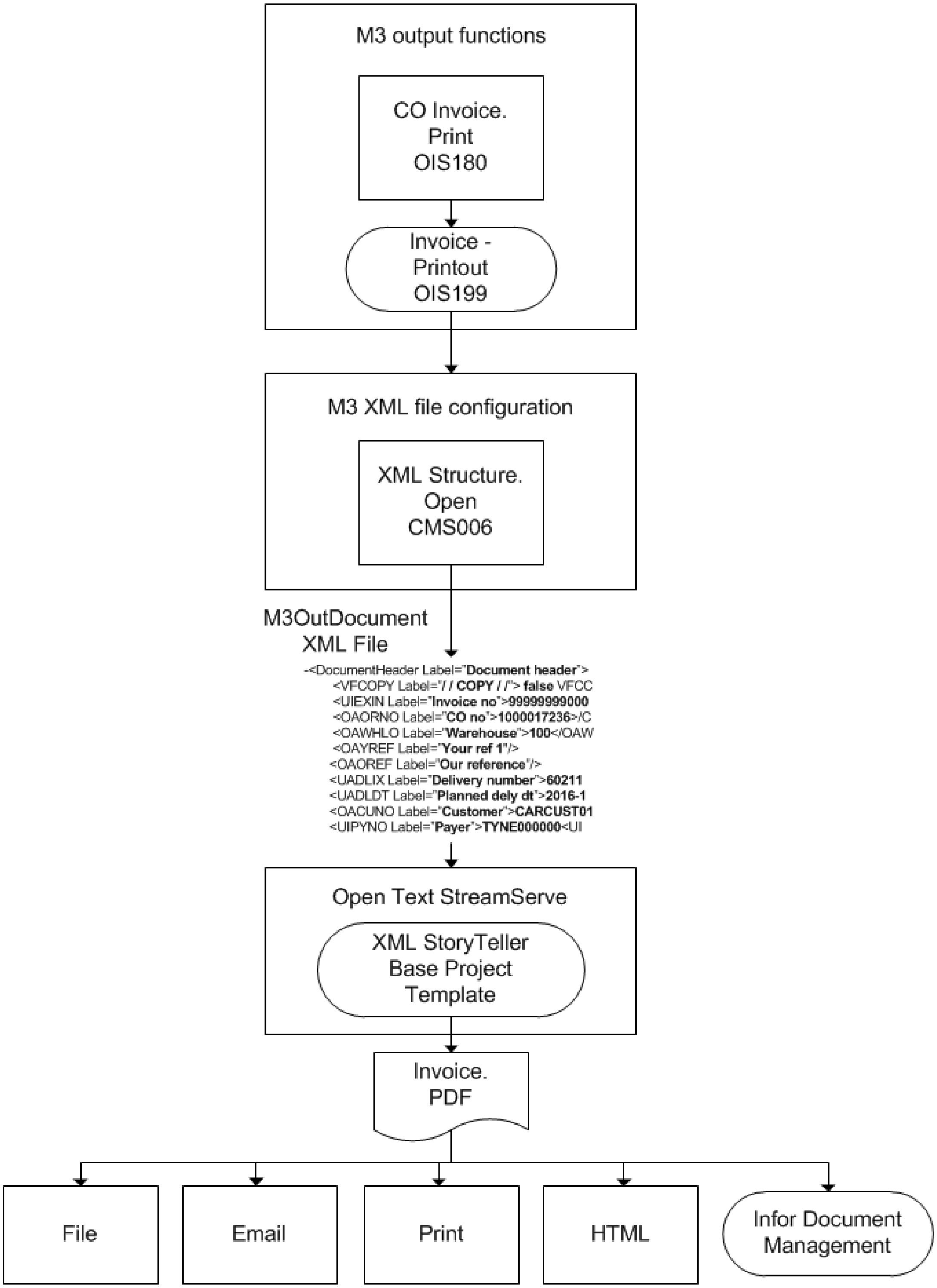
Configuration guidelines

This configuration guide consists of these parts:

- M3 Business Engine data setup and configuration
- M3 Business Engine XML file configuration
- Design and structure of M3OutDocument XML file
- XML StoryTeller base project template for OpenText StreamServe or OpenText Exstream
- Configuring Infor Document Management or IDM for M3 standard output
- Optional: Import configuration data for Russia documents

M3 BE COM process overview

This figure shows an overview of the M3 BE COM process.


M3 output functions
CO Invoice. Print OIS180
Invoice - Printout OIS199
M3 XML file configuration
XML Structure. Open CMS006
M3OutDocument XML File
Open Text StreamServe
XML StoryTeller Base Project Template
Invoice. PDF
File
Email
Print
HTML
Infor Document Management

```
-<DocumentHeader Label="Document header">
     <VFCOPY Label="/ / COPY / /"> false VFCC
     <UIEXIN Label="Invoice no">99999999000
     <OAORNO Label="CO no">1000017236>/C
     <OAWHLO Label="Warehouse">100</OAW
     <OAYREF Label="Your ref 1"/>
    <OAOREF Label="Our reference"/>
    <UADLIX Label="Delivery number">60211
    <UADLDT Label="Planned dely dt">2016-1
    <OACUNO Label="Customer">CARCUST01
    <UIPYNO Label="Payer">TYNE000000<UI
```

Note: OIS180 and OIS199 are only example programs, the program names may vary.

- **M3 output functions**

Existing M3 output functions are reused to generate XML files as an alternative to Streamfiles.

- **M3 XML file configuration**

  To configure the XML file, all metadata for the structure of the XML file are saved in M3 BE tables. Fields and labels can be added using ordinary M3 Business Engine functions, without the need for any Java coding ID.

- **M3OutDocument XML file**

  The output is a highly standardized, multi-level XML file which includes translations of labels, and metadata.

- **Open Text StreamServe**

  The XML file is forwarded to OpenText StreamServe, a third party product. The new layout templates are based on StreamServe StoryTeller. StoryTeller is the replacement of PageOut and it serves as the base for all feature StreamServe enhancements. License for StoryTeller is now included in the M3 Base SKU.

  With the functionalities in StoryTeller, the number of projects is significantly decreased. StoryTeller templates were developed in close collaboration with OpenText StreamServe.

  These are the media types that can be managed by the new StoryTeller templates:

  - Print the document
  - Receive a PDF copy of the document through email
  - Receive an HTML version of the document through email
  - File the document
  - Send the document to IDM

Infor Document Management (IDM)

IDM is Infor’s document management solution which provides a central repository for all business documents across M3 applications.

COM has a built-in connection to IDM using IDM's Application Programming Interface or IDM-API. The mapping of the XML to the document type and attributes is done using ordinary M3 functions, without the need for additional Java coding.

From the IDM client, you can search for documents based on type, attribute, or operation. These are the document management options after obtaining the search results:

- Display
- Check out
- Download
- Delete
- Share document as an Infor Ming.le™ post

- Share

M3 Business Engine data setup and configuration

3

With Configurable Output Management, users can generate configurable XML files, customize documents, and manage them according to their needs.

Based on existing printer functions, configurable XML files can be generated as an alternative to Streamfiles. To determine if a Streamfile or a configurable XML is generated, you can check the settings based on these programs:

- MNS204 - Printer file and/or user
- CRS945 - Control object

Basic configuration steps

1 Start the program "List and Printer Programs. Configure" (CMS005). From this program, all metadata can be generated for all printer functions that support configurable XMLs.

- In the Select field, select 3-Printer file.
- Click ACTIONS, then select "Gen standard" (F14). Click OK. This will create the standard configurations.

  Note: If the printer file is already on the list, use Related option 25 - Reset to standard, before working with the printer file. This will ensure that the settings are updated to the latest standards.

2 Start MNS217. This action defines the interface for configurable XML by creating an output service type in the "Output Service Type. Open" program.

Create a new Service type. On the E-panel, select XML under Interface.

3 Start MNS216. This action defines the service for configurable XML by creating a service identity in the "Output Service. Open" program.

Create a new Service identity. On the E-panel, select the Output Service type created in MNS217. Specify the Server ID and port where layouts are installed. See "Overview" on page 33.

4 Start MNS204. This action connects the service identity for XML to the printer file and user in the "Output Service Selection. Open" program.

5 Start MNS205. This action connects output mediums to the printer file and user in the "Output Media Selection. Open" program.

This table shows the complete list of media options to be used.

See "StreamServe project structure" on page 35 for a list of media options and file types supported by XML StoryTeller base project template for OpenText StreamServe.

Media	Output
*PRT	Printer
*FILE	File
*MAIL	Email
*ARCHIVE	Archive

a To send the output to printer:

Specify *PRT in Media. On the E-panel, specify the detailed information for the printer. Define the printer address in the program "Printer. Open" (CRS290).

b To file the output:

Specify *FILE in Media. On the H-panel, specify the detailed information for the destination folder.

c To receive an email of the output:

Specify *MAIL in Media. On the F-panel, specify the email address.

d To archive the output:

Specify *ARCHIVE in Media. No additional settings are needed.

6 Optionally, you may want to control the media settings using control objects. Use program CRS949 as an alternative to MNS204 and MNS205.

Specify CRS945, which opens the "Std Document. Connect Media Ctrl Object" program. If this is used for a specific printer file, the interface is defined in "Document Media. Open" (CRS116). Then, it is connected to the standard document defined in "M3 Document. Connect Media" (CRS929).

7 Start CRS010. This step will define the ISO language code for all languages in the "Language. Open" program. Choose your preferred language option. Suppose you choose English. The Lng code is GB while the ISO code is en. The ISO language code is used when setting the field to "local" in the XML file.

See section on "Formatting numbers and dates" on page 38 to learn how the Local field is used in the XML StoryTeller Base Project Template for OpenText StreamServe.

8 To create an XML file, run any Printer function that is listed in the program CMS005. Ensure that the Service identity from MNS216 is used.

Follow this procedure to view the XML file:

a Start MNS206. This action opens the "Output. Manage per Job" program.

b Select the Job number of the output where you would like to view the XML file. Right-click Related, then click View (CTRL+11). The M3 Browse window is displayed.

c Highlight the Job number, then click Select. A pop-up window with the XML page is displayed. You can use the options Download or Show XML.

Note: Ensure that the pop-up blocker is disabled on the browser. If not, move over your pointer to the address bar, then click the pop-up blocked icon. If a message is displayed, select the first option for allowing pop-ups.

Variants

Variants are used to separate different types of documents which are generated from the same printer function. For example, the print function "Customer Invoice. Enter Manual" (ARS120) generates a typical invoice, a credit note, or a corrective invoice. The variant has a hard connection to the Report Layout functionality.

Example

Variant	Description
1	Invoice
2	Credit note
3	Corrective invoice

Variants are based on the printer file and system defined. These cannot be added, changed, or deleted. Variants per printer file are displayed in the program "Printer Files XML Variants. Open" (CMS004).

Report layout

Report Layout is used to define the structure of the XML file when running a printer function. This is done by connecting Report Layout to a specific XML structure in the program "Report Layout. Open" (CMS025).

Then, the report layout works in two ways:

1. The Report Layout is manually selected when running the printer function, usually for reports.
2. The Report Layout is connected to the print function in the program "Report Layout Selection. Open" MNS210, usually for documents.

Example

Report layout	Description	Variant	XML structure
M3_STD_01-01	Invoice	1	M3_STD_01-01
M3_STD_02-01	Credit note	2	M3_STD_02-01
M3_STD_03-01	Corrective invoice	3	M3_STD_03-01

By choosing different report layouts, it is possible to create different kinds of XML files from the same printer file.

One M3 BE Standard Report Layout is delivered for every printer file and variant. The name always starts with M3 and Standard Report Layouts cannot be deleted.

M3 Business Engine XML file configuration

4

To configure the XML file, all metadata for the XML file is saved in normal BE tables. Infor delivers a standard configuration including a subset of fields based on the standard tables used in the print function.

Note: Standard tables are tables which are always read by the Print function.

You can make additional fields available for the Print function. Any field in any BE table can be made available for the Print function using ordinary M3 Business Engine functions, without the need for Java coding. Virtual fields can also be created. After a field has been made available for the Print function, it can be added to the XML file.

These are the ways to make additional fields available for the Print function:

- For fields from standard tables, use the option "Add field to field group" (CTRL+7) in the "XML Section. Open Table" (CMS008) program.
- For tables with a one-to-one relation against a standard table, the fields can be made available by defining a table in the "Related Tables. Connect" (CMS011) program.
- For tables with a one-to-many relation against a standard table, the fields can be made available by defining an XML section in the "XML Section. Tables" (CMS008) program.
- Use the "Virtual Fields. Open" (CMS012) program to define virtual fields.

XML section tables

In the "XML Section. Open Table" (CMS008) program, all tables for each XML section are listed. To start CMS008, use Related "XML section tables" (CTRL+13) in the "List and Printer programs. Configure" (CMS005) program. You can also use the "Section tables" option in the "XML Structure. Open Section" (CMS007) or "XML Structure Section. Open Element" (CMS009) programs.

In CMS008, additional fields from tables can be made available and new sections can be created. After creating a section in CMS008, the section can be added to a specific XML structure in the "XML Structure. Open Section" (CMS007) program as a custom section.

To make additional fields from a table available in CMS008/B, follow these steps:

1 Click Options, then click "Add to field group" (CTRL+7).

2 On the N-panel, specify the Field selection. This determines if all fields or only specific fields from the table is made available. For the second option, a maximum of 10 fields can be created at the same time.

3 Press Enter to add the field. After the fields are created, each field can be selected in an element from the CMS009 program.

To create a section in CMS008/B, follow these steps:

1 Specify the XML section and the table to be read in the section. The value of the XML section must be larger than 100. Click CTRL+1 (Create).

2 On the E-panel, the Field prefix is prepopulated and is normally not changed. Specify the Base XML section.

3 On the F-panel, specify the Sorting option used to read the related table. Specify the Read option for every Key field in the table based on the Sorting option selected. To read multiple records in the table, option 9 "Read all records" is used.

4 On the N-panel, specify the Field selection. This determines if all fields or only specific fields from the table is made available. For the second option, a maximum of 10 fields can be created at the same time.

5 Press Enter to create the section. If additional fields must be created, click CTRL+7 from CMS008/B. After the section is created, the section can be added as a custom section in CMS007 using Related "Custom sections" (CTRL+21).

Related tables

For tables with a one-to-one relation against a standard table, the fields can be made available for the Print function by defining a related table in the "Related Tables. Connect" (CMS011) program.

To start CMS011, click Related, then click Related tables (CTRL+11) in the "List and Printer programs. Configure" (CMS005) program. You can also use the "Related table" option in the "XML Structure Section. Open Element" (CMS009) program.

In "Related Tables. Connect" (CMS011), up to 30 related tables can be defined.

To create a related table in CMS011/B, follow these steps:

1 Specify the table and click CTRL+1 (Create).

2 On the E-panel, specify the Sequence number which indicates the read sequence used for the related table.

3 On the F-panel, specify the Sorting option used to read the related table. Specify the Read option for every Key field in the related table, based on the Sorting option selected.

4 On the N-panel, specify the Field selection, which determines if all fields or only specific fields must be created. For the second option, a maximum of ten fields can be created at the same time.

5 Press Enter to activate the related table and create the selected fields. If additional fields must be created, click CTRL+7 to add items to field group from the CMS011/B program.

After the fields are created, the fields can be selected in an element from the CMS009 program.

Virtual fields

Fields that do not exist in any table can be defined as virtual fields in the "Virtual Fields. Connect" (CMS012) program. To start CMS012, click Related, then click Virtual fields (CTRL+12) in the "List and Printer programs. Configure" (CMS005) program. You can also use the "Virtual fields" option in the "XML Structure Section. Open" (CMS009) program.

In CMS012, these types of virtual fields can be defined:

- Calculated fields
- Description or Name from the Related system table
- Currency conversion
- Logical formula
- Date conversion
- Merge data

An unlimited number of virtual fields can be defined. After creating the virtual fields, the fields can be selected in an element from the CMS009 program.

XML structure

The XML structure defines the element tree. An element tree always starts at a root element and branches out from the root to child elements. To configure the XML structure, start the "XML Structure. Open" (CMS006) program.

These are the two formats available.

1 **M3 standard format**

This format is used for templates and standard layouts. To ensure that the layout is working, there are restrictions when it comes to what a user can change.

The root element and structure or sections of the XML file are predefined and cannot be changed. The user can extend the XML file by adding elements within the custom section. It is also possible to override the standard labels.

One M3 BE standard XML structure is delivered for every printer file and variant. The name always starts with M3 and it cannot be deleted.

Section name	Element name	Number of elements
Fixed	Fixed	Unlimited

2 **Open format**

This format is used for other types of XML files or when the application which processes the XML requires a specific format. With open format, there are no restrictions as to what a user can change.

Section name	Element name	Number of elements
Free	Free	Unlimited

The XML output can be customized further using split XML file. Select the check box to split multiple printouts into separate XML files.

XML sections

The XML structure is divided into logical parts which take the form of XML sections. Each XML section is configured in the "XML Structure. Open Sections" (CMS007) program. To start CMS007, use the Related option in "XML Structure. Open Section" (CTRL+11) in the "XML Structure. Open" (CMS006) program.

There are different predefined XML sections for each printer function and variant. These predefined XML sections are automatically created when a new XML structure is created. By setting the status to 10 on a section, it will not be included in the XML file. Predefined XML sections cannot be deleted.

Custom XML sections can be created using the Related option "Custom sections" (CTRL+21).

Note: A section must be created first in the CMS008 program, with its corresponding Base section.

If new, predefined sections are added by Infor, it can be added to existing structures using Related option 20 - "Update XML section" from CMS006.

XML elements

The XML elements included in a specific section is defined in the "XML Structure Section. Open Element" (CMS009) program. To start CMS009, use the Related option "XML Structure. Open Section" (CTRL+11) in the "XML Structure. Open Sections" (CMS007) program.

Element type

Element type	Description
1	Text element
2	Start element
3	End element

Modifying XML elements

For M3 standard format, the XML elements are divided into three groups based on the element sequence number. This table shows the purpose of each group:

1. **M3 standard format**

   For M3 standard format, the XML elements are divided into three groups based on the element sequence number. This table shows the purpose of each group:

Element sequence number	Purpose	Modifications	Restrictions
0001-3999	These predefined standard elements are only used for M3 BE standard XML structures.	Labels	Adding or deleting elements in the group
4001-6998	These predefined standard country elements are only used for M3 BE standard XML structures when a table is located in a country configuration component.	Labels	Adding or deleting elements in the group
7001-9998	These are custom elements.	Elements can be freely added, changed, or deleted	

2. **Open format**

   For open format, XML elements can be freely added.

Source of data

Element type 1 contains the text or data in the XML file. The field Source of data specifies which data to include. These are the most common sources of data:

Field value	This field is used when fetching data from other tables. If the field does not exist as a standard field, it can be added by connecting the field to related tables.
Specific value	This field is used to specify any value that can be used as static information.

Design and structure of M3OutDocument XML file

5

All M3OutDocument XML files follow a strict standard, regardless of the type of document. The description of XML file structure provides users a better understanding of XML structures in general.

```
<M3OutDocument> Mandatory. Always single occurrence
                <Metadata> Mandatory. Always single occurrence
                <DataArea> Mandatory. Always single occurrence
                        <Document> Mandatory. Single or multiple occurrence
                        <Document> Mandatory. Single or multiple occurrence
                </DataArea>
</M3OutDocument>
```

- **M3OutDocuments**

  The root node is always M3OutDocument

- **Metadata**

  Provides general information about the XML file

- **DataArea**

  Includes all data in the XML file

  The DataArea can include one Document or more.

- **Document**

  Can be single or multiple occurrence

  One Document can represent an invoice, a purchase order, or a statement of account number. A report normally includes just one Document.

- **M3StandardFormatVersion**

  Is always set to 1

- **ReportLayout**

  Selects report layouts

- **Structure**

  Defines XML structure

  This is connected to the "Report Layout. Open" program in CMS025.

- **DocumentDivision**

Defines the division for Document, not to be mistaken with division for user-running function

- **CountryVersion**

  Defines the country version through the "Company Connect. Division" program in MNS100

- **BaseCountry**

  Defines the country of the Document division generating the document

  The only exception is if fiscal representation is used.

- **FromToCountry**

  Defines the country code for the receiver of the document through the "Country. Open" program in CRS045

- **DocumentLanguage**

  Defines the language code for the receiver of the document through the "Language. Open" program in CRS010

  DocumentLanguage input is based on the user, customer, or supplier.

- **Locale**

  Defines the ISO codes used

  Locale uses the format xx-yy which represents these details:

  - XX - ISO code for the language used based on DocumentLanguage. This is defined in the "Language. Open" program in CRS010.
  - YY – ISO code for the country based on FromToCountry. This is defined in the "Country. Open" program in CRS045.

Language		**Country**		**Locale**
M3 code	ISO code	M3 code	ISO code	
GB	en	US	US	en-US
GB	en	GB	GB	en-GB
DE	de	DE	DE	de-DE
SE	sv	SE	SE	sv-SE

- **DateFormat**

  Defines the document date based on the data required by the user, customer, or supplier

- **DecimalFormat**

  Defines the decimal format for the document based on the data required by the user, customer, or supplier

- **ThousandSeparator**

Is always set opposite the separator from DecimalFormat

Document

This is an example of how the Document section looks like:

```
       <Document>
               <Cover> Optional. Selection and filter fields. Always single occurrence.
               <Formatting/> Mandatory. Information of how data should be formatted.
               <Header/> Mandatory. Information for the top of the printout page.
               <DocumentHeader/> Optional. Document header information.
               <TextBlock/> Optional.
               <SubDocument/> Mandatory. If document is split into sub-documents. Normally single
                     <TextBlock/> Optional.
                     <Lines/> Optional. Groups all lines.
                     <TextBlock /> Optional.
               <TextBlock/> Optional.
               <VATSummaryBox/> Optional.
               <DocumentTotal/> Optional.
               <Media/> Mandatory. Information for distribution of the printout.
       </Document>
```

Standard sections

Formatting

Includes metadata to help the user format the layout

```
<Formatting>
        <M3StandardFormatVersion>1</M3StandardFormatVersion>
        <ReportLayout>M3_STD_03-01</ReportLayout>
        <Structure>M3_STD_03-01</Structure>
        <PaperSize>A4</PaperSize>
        <Localization>
            <DocumentDivision>E10</DocumentDivision>
            <CountryVersion>SE</CountryVersion>
            <BaseCountry>SE</BaseCountry>
            <FromToCountry>GB</FromToCountry>
            <DocumentLanguage>GB</DocumentLanguage>
            <Locale>en-GB</Locale>
            <DateFormat>YYMMDD</DateFormat>
            <DecimalFormat>,</DecimalFormat>
            <ThousandSeparator>.</ThousandSeparator>
         </Localization>
         <DocumentInformation>
</Formatting
```

Data element

Unique identifier and labels

The unique identifier for a text element is the element name. The element name is normally equal to the field name in the M3 BE database.

Field label is included in the Document language as attribute "Label".

```
<UIIVNO Label="Invoice no">201401561</UIIVNO>
<OAORNO Label="CO no">1000025077</OAORNO>
<OAWHLO Label="Warehouse">110</OAWHLO>
```

Decimal data type

Data under the decimal data type is included in the XML according to these rules:

- Without ThousandSeparator
- No zero suppress (0.00)
- Decimal format is a period (.)
- Figure is rounded to the correct number of decimals and adjusted according to price quantity

Metadata for how a decimal field should be formatted is included in the section "Localization".

```
<Formatting>
…
<Localization>
         <DecimalFormat>,</DecimalFormat>
         <ThousandSeparator>.<ThousandSeparator>
         <Local>en-GB<Local>
</ Localization >
```

If Debit/Credit code should be used, it is indicated in the element DebitCreditCodeUsed (true/false), section DocumentInformation.

```
<Formatting>
…
<DocumentInformation>
       <DebitCreditCodeUsed>true</DebitCreditCodeUsed>
</DocumentInformation>
```

The currency amount fields that use Debit/Credit code has an additional attribute "D-C", including the Debit/Credit code sign.

```
<VFORTO Label="Order total" D-C="D">68325.00</VFORTO>
<VFTOPY Label="To pay" D-C="D">68325.00</VFTOPY>
```

Date data type

Data under the date data type is shown using the format YYYY-MM-DD with these details:

- [YYYY] indicates a four-digit year, [MM] indicates a two-digit month, [DD] indicates a two-digit day.
- Each date value has a fixed number of digits that is padded with leading zeros.
- The separator used between date values is the hyphen (-).

XML example without relation to time zone

Example: 14:05:23

```
<UHTIID Label="Invoice time">14:05:23</UHTIID>
```

XML example with relation to time zone

If the date has a relation to a time zone that is offset from Coordinated Universal Time or UTC time, the number is added as a positive or a negative value behind the date. There is no need to use the UTC time offset when specifying the date on a document. It is only used if the date needs to be converted to UTC time.

```
<UHIVDT Label="Invoice date">2016-03-21+5:00</UHIVDT>
```

Metadata for how a date field should be formatted is included in Section "Localization".

```
<Formatting>
…
<Localization>
         <DateFormat>YYMMDD</DateFormat>
         <DateEditing>/<DateEditing>
         <Local>en-GB<Local>
</ Localization >
```

M3 setting for date format	Text in element DateFormat
YMD	YYMMDD
DMY	DDMMY
MDY	MMDDYY
YWD	YYWWD

Time data type

Data under the time data type is shown using the format hh:mm:ss with these details:

- [hh] indicates the hour, [mm] indicates the minutes, [ss] indicates the seconds.
- Each time value has a fixed number of digits that is padded with leading zeros.
- The separator used between date values is (:)

XML example without relation to time zone

Example: 14:05:23

```
<UHTIID Label="Invoice time">14:05:23</UHTIID>
```

XML example with relation to time zone

If the date has a relation to a time zone that is offset from Coordinated Universal Time or UTC time, the number is added as a positive or a negative value behind the date. There is no need to use the UTC time offset when specifying the date on a document. It is only used if the date needs to be converted to UTC time.

```
<UHTIID Label="Invoice time">14:05:23+5:00</UHTIID>
```

Standard sections

DocumentHeader

The section DocumentHeader contains information that is common for the Document. The Address section is placed within the Document.

XML example with one address

```
<DocumentHeader Label="Document header">
         <UIIVNO Label="Invoice no">201401561</UIIVNO>
         <OAORNO Label="CO no">1000025077</OAORNO>
         <OAWHLO Label="Warehouse">110</OAWHLO>
         <Address Label="Delivery Address Type="Delivery">
</DocumentHeader>
```

XML example with two addresses

```
<DocumentHeader Label="Document header">
         <UIIVNO Label="Invoice no">201401561</UIIVNO>
         <OAORNO Label="CO no">1000025077</OAORNO>
         <OAWHLO Label="Warehouse">110</OAWHLO>
         <Address Label="Delivery Address Type="Delivery">
         <Address Label="Payer Address Type="Payer">
</DocumentHeader>
```

Addresses

Addresses are separated by type. Type is included as an attribute and can have these values:

Type	Description
Delivery	Delivery address
Customer	Customer address
Payer	Payer address
Invoice	Invoice address

Type	Description
Supplier	Supplier address
Payee	Payee address
Company	Company address
Location	Location address
Quotation	Quotation address
Bank	Bank address

These elements are always included in the Address section:

- CUNM - Name
- ADR1 - Address line 1
- ADR2 - Address line 2
- ADR3 - Address line 3
- ADR4 - Address line 4

XML example with one address

```
<Address Label="Delivery Address Type="Delivery">
        <VFCUNM>Infor company</ VFCUNM >
        <VFADR1>Utterstreet</VFADR1>
        < VFADR2>Number 4</VFADR2>
        < VFADR3>187 Taby</VFADR3>
        < VFADR4>Sweden</VFADR4>
</Address>
```

XML example with two addresses

```
<Address Label="Delivery Address Type="Delivery">
        <VFCUNM>Infor company</ VFCUNM >
        <VFADR1>Utterstreet</VFADR1>
        < VFADR2>Number 4</VFADR2>
        < VFADR3>187 Taby</VFADR3>
        < VFADR4>Sweden</VFADR4>
</Address>
<Address Label="Payer Address Type="Payer">
        <VFCUNM>Infor Sweden AB</ VFCUNM >
        <VFADR1>Borgarfjordsgatan 9</VFADR1>
        < VFADR2></VFADR2>
        < VFADR3>164 40 Kista</VFADR3>
        < VFADR4>Sweden</VFADR4>
</Address>
```

Text block

Text block is separated by types. Type is included as an attribute and can have these values:

Type	Subtype
ExemptionText	
DocumentText	PreText or PostText
CustomerText	PreText or PostText
OrderText	PreText or PostText
OrderLineText	PreText or PostText
ItemText	PreText or PostText
CoreTerms	
ConfigText	
AssignmentText	PreText or PostText
AgreementText	PreText or PostText
GeneralText	
AttributeText	
FinReasonText	

XML example with one text block

```
<TextBlock Label="Document text" Type="DocumentText" SubType="PreText">
       <BD-3>asldfjasljfalsjfaslfdj</BD-3>
       <BD-3>hhdftdgsdfsfsfs</BD-3>
       <BD-3>sdwe63gdfg34te</BD-3>
       <BD-3>xxrwrwtey567</BD-3>
……
</TextBlock>
```

XML example with two text blocks

```
<TextBlock Label="Document text" Type="DocumentText" SubType="PreText">
       <BD-3>asldfjasljfalsjfaslfdj</BD-3>
       <BD-3>hhdftdgsdfsfsfs</BD-3>
       <BD-3>sdwe63gdfg34te</BD-3>
       <BD-3>xxrwrwtey567</BD-3>
……
</TextBlock>
<TextBlock Label="Order text" Type="OrderText" SubType="PreText">
       < BD-3>ddddfafj</BD-3>
       < BD-3>dddddddddsss</BD-3>
       < BD-3>4353333333</BD-3>
</TextBlock>
```

Lines

Line is repeating and is grouped by Lines. Labels are included only once before any Line section.

In many cases, SublineValues are included at the break of key fields as subtotal values and are used for sorting.

XML example with no subtotal values

This example shows a Lines Label section with only a list of one or many lines.

```
<Lines Label="Lines">
         <UBPONR Label="Line"/>
         <UBITNO Label="Item number"/>
         <Line>
               <UBPONR>1</UBPONR>
               <UBITNO>MB_101</UBITNO>
         </Line>
         <Line>
               <UBPONR>2</UBPONR>
               <UBITNO>MB_202</UBITNO>
         </Line>
         <Line>
               <UBPONR>3</UBPONR>
               <UBITNO>MB_303</UBITNO>
         </Line>
 </Lines>
```

XML example with subtotal values

This example shows a Lines Label section with subtotal values or SublineValues included:

```
<Lines Label="Lines">
        <UBPONR Label="Line"/>
        <UBITNO Label="Item number"/>
        <Line>
              <UBPONR>1</UBPONR>
              <UBITNO>MB_101</UBITNO>
        </Line>
        <Line>
              <UBPONR>2</UBPONR>
              <UBITNO>MB_202</UBITNO>
        <SublineValues>
              <Element 1/>
              <Element 2/>
       </SublineValues>
       <Line>
              <UBPONR>3</UBPONR>
              <UBITNO>MB_303</UBITNO>
       </Line>
       <Line>
              <UBPONR>4</UBPONR>
              <UBITNO>MB_404</UBITNO>
       <SublineValues>
              <Element 1/>
              <Element 2/>
       </SublineValues>
</Lines>
```

XML example with sorting by address

If lines are grouped with subtotals, Subline must be used. This example shows a Lines Label section where the address information is included for each Subline:

```
<Lines Label="Lines">
        <UBPONR Label="Line"/>
        <UBITNO Label="Item number"/>
        <Subline>
               <Address Label="Delivery Address Type="Delivery">
               <Line>
                      <UBPONR>3</UBPONR>
                      <UBITNO>MB_303</UBITNO>
               </Line>
        </Subline>
        <Subline>
               <Address Label="Delivery Address Type="Delivery">
               <Line>
                     <UBPONR>4</UBPONR>
                     <UBITNO>MB_404</UBITNO>
              </Line>
              <Line>
                    <UBPONR>5</UBPONR>
                    <UBITNO>MB_505</UBITNO>
              </Line>
        </Subline>
</Lines>
```

XML example with pre- and post- text

If pre- and post-text is used for the Lines Label section, they are placed within the Line.

```
<Lines Label="Lines">
         <UBPONR Label="Line"/>
         <UBITNO Label="Item number"/>
        <<Line>
               <UBPONR>3</UBPONR>
               <UBITNO>MB_303</UBITNO>
               <TextBlock Label="Item text" Type="ItemText" SubType="PreText">
               <TextBlock Label="Item text" Type="ItemText" SubType="PostText">
        </Line>
</Lines>
```

VATSummaryBox

For the VATSummaryBox, Level 03 is always used. Even if not all fields are used, all fields will be defined, as shown in this table:

Amount	VAT	VAT 1(%)	VAT 2(%)	Original invoice	This invoice	Net amount
Base	&VCTD			&VBAO	&VBAT	&VBAN
VAT	&VCTD	&VTP1		&VAO1	&V	
VAT						
&LOCU						

XML example:

```
<VATSummaryBox>
      <VFVTCD Label="Vcd"/>
      <VFVT40 Label="Description"/>
      <VFVTP1 Label="VAT 1"/>
      <VFVBAT Label="This invoice"/>
      <VFVBAO Label="Orig invoice"/>
      <VFVBAN Label="Net"/>
      <VFVAT1 Label="This invoice"/>
      <VFVAO1 Label="Orig invoice"/>
      <VFVAN1 Label="Net"/>
      <VFVTP2 Label="VAT 2"/>
      <VFVAT2 Label="This invoice"/>
      <VFVAO2 Label="Orig invoice"/>
      <VFVAN2 Label="Net"/>
      <VFLOCU Label="Local currency"/>
      <VFLOVB Label="Base amount"/>
      <VFLOVA/>
      <VFLORA Label="Rate"/>
      <VAT Label="VAT">
             <VFVTCD>1</VFVTCD>
             <VFVT40>Vat code 20%</VFVT40>
             <VFVTP1>10.00</VFVTP1>
             <VFVBAT>10640.00</VFVBAT>
             <VFVBAO>0.00</VFVBAO>
             <VFVBAN>10640.00</VFVBAN>
             <VFVAT1>1064.00</VFVAT1>
             <VFVAO1>0.00</VFVAO1>
             <VFVAN1>1064.00</VFVAN1>
             <VFVTP2>0.00</VFVTP2>
             <VFVAT2>0.00</VFVAT2>
             <VFVAO2>0.00</VFVAO2>
             <VFVAN2>0.00</VFVAN2>
             <VFLOCU>SEK</VFLOCU>
             <VFLOVB>21280.00</VFLOVB>
             <VFLOVA>2128.00</VFLOVA>
             <VFLORA>2.00</VFLORA>
       </VAT>
</VATSummaryBox>
```

HeadCharges

For the HeadCharge section, the charge is repeated and grouped by HeadCharges. The labels are only included once before any Charge section.

XML example with one charge

```
<HeadCharges>
       <USCRD0 Label="Description"/>
       <USCRID Label="Charge"/>
       <VFCHAM Label="Charge amount"/>
       <VFVTCD Label="Vcd"/>
       <Charge Label="Charge">
               <USCRD0>Monica's Head charge_VATcd01</USCRD0>
               <USCRID>MBHC1</USCRID>
               <VFCHAM>200.00</VFCHAM>
               <VFVTCD>1</VFVTCD>
       </Charge>
</HeadCharges>
```

XML example with two charges

```
<HeadCharges>
      <USCRD0 Label="Description"/>
      <USCRID Label="Charge"/>
      <VFCHAM Label="Charge amount"/>
      <VFVTCD Label="Vcd"/>
      <Charge Label="Charge">
              <USCRD0>Monica's Head charge_VATcd01</USCRD0>
              <USCRID>MBHC1</USCRID>
              <VFCHAM>200.00</VFCHAM>
              <VFVTCD>1</VFVTCD>
      </Charge>
      <Charge Label="Charge">
              <USCRD0>Monica's Head charge_VATcd03</USCRD0>
              <USCRID>MBHC3</USCRID>
              <VFCHAM>400.00</VFCHAM>
              <VFVTCD>1</VFVTCD>
      </Charge>
</HeadCharges>
```

DocumentTotal

For DocumentTotal, Level03 is always used. It will contain all elements adding the total for the whole Document.

Additional labels

Labels not related to any data are included as normal labels, but with the element value set to true or false. Additional labels are always included first in a section.

Example 1

Text "***PRELIMINARY***" included in the DocumentHead

*****PRELIMINARY*****

XML example

```
<DocumentHead Label="Document header >
               <VFPREL Label="Preliminary">true</VFRELI>
…..
</DocumentHead>
```

Example 2

Signature line with "Place", "Date", and "Signature" in DocumentTotal

Place Date Signature

XML example

```
<DocumentTotal Label="Document total">
       <VFPLAC Label="Place">true</VFPLAC>
       <VFDATE Label="Date">true</VFDATE>
       <VFSIGN Label="Signature">true</VFSIGN>
…..
</DocumentTotal>
```

XML StoryTeller project templates for OpenText StreamServe

6

Overview

OpenText StreamServe is a third party application used to transform M3 Business Engine data into predefined documents or report layouts. StreamServe allows many different formats and distribution channels for data including email, mobile, electronic invoice, and printer. Infor has produced an XML StoryTeller Base Project Template for StreamServe. The StreamServe application makes modification and adaptation of the template layouts easier.

Component	Supported version
OpenText StreamServe	5.6.2
OpenText Exstream	16.4

For StreamServe, these are the components delivered:

- Main projects to create a runtime environment
- All projects, source files for the layouts, test data, and example reports to be used for customer modifications
- Two separate dcpackage files in the base package for the M3 Business Engine

  Note: The dcpackage format is an internal format for StreamServe.

License file

The XML StoryTeller Base Project Template uses StreamServe SKUs not previously included in the M3 base SKU. If a license file is generated before October 1, 2016, a new license file must be requested from OpenText. It will then include the additional keys for the added SKUs.

Main projects structure

Two main projects are delivered:

- M3_BE_15.1.4_XML_Base_Project_Template_vxx.zip, referred to as the Base project for the rest of this document.
- M3_BE_15.1.4_XML_Forms_Project_Template_vxx.zip, referred to as the Forms project for the rest of this document.

The main difference between these projects is that in the Base project, language translations are sent in the XML while in the Forms project, the translation tables are installed in StreamServe. Another difference between the two is that layouts are more defined in Forms project, and are specific for some country versions.

Adding a project to a StreamServe installation

1. Open and unpack the zip file M3_BE_15.1.4_XML_Base_Project_Template_vxx.zip.
2. Start OpenText Design Center. Click File > Unpack Project, then select the unpacked file M3_BE_15.1.4_XML_StoryTeller_Base_Project_Template_vxx.dcpackage. Select a destination folder. Ensure that "Keep folder structure" and "Open project after unpacking" are selected.
3. If you did not open the project after extraction, open the InforM3MainProject.dcproject file from the MainProject folder.
4. In Design Center, select Platform. Modify the input and output connectors according to your needs. See "Platform - Input/Output connectors" on page 36.
5. Save the project.
6. Click Tools, then click Export. Take note of the Export directory.
7. Open Control Center and create a new application. For example, the new application can be named M3XML_DEV. The suffix refers to the physical layer deployed to the application. You must have at least one application for each physical layer used.

   For detailed instructions on how to create a new application, visit http://www.streamserve.com for the StreamServe User Documentation.
8. Java configuration must be enabled on the application because of the IDM output connector. Right-click the application in Control Center and select Java configuration. Click the field Value under Java vendor. From the drop-down, select Oracle or IBM.
9. Right-click the application and select "Deploy Export File". Then, highlight the exported file created. Choose one physical layer to deploy. For example, DEV can be deployed for an application named M3XML_DEV. Start the application.
10. To test the application without using M3:
    - Open the project with Design Center.
    - Open the InvoiceProject resource folder.
    - Right-click OIS199-Sample under the Samples folder. Select Extract to File and then select a destination.

- Open OIS199-Sample.xml. Find and replace john.doe@mailserver.com with your email address.
- Copy the file and paste in the location of the input folder.
- The file is processed and the user will receive an email.

11 If required, repeat these steps for the Forms project.

StreamServe project structure

The StreamServe Base project is divided into 18 different sub-projects, which are all connected to the main project.

1 InvoiceProject

2 ReportProject

3 EQM - Equipment Quotation Management Project

4 FIC - Financial Controlling

5 FIM - Financial Accounting

6 MAI - Maintenance Management

7 MAN - Manufacturing Execution

8 MNT - Maintenance Order Management

9 MSF - Application Foundation

10 PJM - Project Management

11 PMD - Product Data Management

12 PUR - Procurement

13 RTM - Rental Management

14 SCE - Supply Chain Execution

15 SCM - Supply Chain Management

16 SCP - Supply Chain Planning

17 SLS - Sales Management

18 SRV - Service Management

Two of these sub-projects, InvoiceProject and ReportProject, aim to merge common documents from multiple business areas.

The StreamServe Forms project has one sub-project, RU-Russian market.

For the StreamServe global platform, there are three physical layers: DEV, TEST, and PROD. These layers allow the user to export files to different environments separately, with the option of having different platform and runtime settings for each physical layer.

Platform - Input/Output connectors

In both the Base and Form projects, a number of input and output connectors are delivered. Some of these connectors must be configured in order to work, and depending on the need. Configuration is done on the platform.

These input connectors are delivered:

1. **TCP/IP input**

   This standard server address is the location where the application is running. Input ports can be modified as needed.

Physical layer	Base project	Forms project
DEV	20150	20160
TEST	20151	20161
PROD	20152	20162

2. **Directory scan**

   In most cases, the input directory location should be modified.

Physical layer	Base project	Forms project
DEV	C:\IO\InforInDEV	C:\IO\InforFormsInDEV
TEST	C:\IO\InforInTEST	C:\IO\InforFormsInTEST
PROD	C:\IO\InforInPROD	C:\IO\InforFormsInPROD

The same output connectors are used in both the Base project and the Forms project. These are the output connectors delivered:

- **MAIL_PDF output**

  For this connector, the TCPIP profile GenericMail is used for mail server settings. The profile must be modified with host and port for your mail server. Profile is located in the Global resource set, under the Profiles folder.

  The connector must also be modified with domain name.

- **MAIL_HTML output**

  This connector has the same configuration as MAIL output.

- **ATTACHMENT_PDF**

  No configuration required, this is used by MAIL output.

- **FILE_PDF output**

This connector does not need to be configured unless used for special functionality.

- **PCL output**

  This connector does not need to be configured unless it will be used for special functionality. But the user has to update the printer information in M3. This can be done in the "Printer. Open" program in CRS290 by adding the printer address in the server address field.

- **MATRIX output**

  This connector has the same setup as PCL, but the user has to update the printer information in M3. This can be done in the "Printer. Open" program in CRS290 by adding MATRIX in the modification field.

- **IDM output**

  This connector requires a path to an IONAPI key. This key can only be generated by a system administrator for a generic user. The key needs to be placed on the same server as the StreamServe application. See "Configuring IDM for M3 standard output" on page 41.

Project runtimes

The runtimes in the base project do not need to be configured unless special functionality is required.

Sample files

Sample XML files are included in each sub-project which makes it easy for the user to get started with StreamServe configuration.

Global function file

No configuration for global functions unless special behavior is required.

StoryTeller fragments

Fragments are a tool to make it easier to maintain documents in StoryTeller. These fragments are general components that can exist in different documents at the same time. One advantage is that they only need to be updated once. In the Base project, some standard fragment documents are provided. A user can then add more fragments to these documents or edit existing fragments, based on the need.

XMLIN event tool

The data from M3 is sent in the form of XML data. To collect the data, XMLIN is required. XMLIN provides a mapping for all standard fields. If a user adds a field to the XML in the M3 environment, the field also has to be added to the XMLIN mapping. Patterns are defined to connect the XMLIN event to specific printer files.

XSLT patterns are used to define fields and patterns in the XMLIN tool.

StoryTeller document

Inside the StoryTeller document, the actual output document is designed.

The main document template design consists of two pages. The first page is required and the succeeding pages are optional. The header data for the document is designed by using various fragments. The part which can grow in size depending on input data is placed into a story, which is then put into a story frame, which is subsequently placed on the page layout.

Most of the components are designed with the help of tables, repeaters, and substitutions. Repeaters and substitutions are controlled by xpath. This makes xpath the main means to control and adapt the layout based on the input data.

These are some useful xpath expressions:

- /data/message/Invoice_No[. != ""])– If used in a repeater on a table row, this expression will not display the row unless an invoice number is sent in the xml.
- string(Line_Amount[. != 0]) – If used in a substitution, this expression will not display the field if the value is zero.
- TextBlock[TextLine] – If used in a repeater on a table, this will not create the textblock unless the TextBlock has a TextLine.

Formatting numbers and dates

All numbers and dates are formatted inside StoryTeller.

The standard delivered formatting is done with the help of variables which can be used in the modification field inside StoryTeller. These variables utilize the locale setting of XML to format numbers correctly, depending on the language and the country. The date format, thousand separator, and decimal format are not used in the standard solution. The value of the locale can be accessed using the **$locale** variable.

To format numbers and dates according to locale using the modification field, these variables need to be set:

- $localeNumCurrFormat

  This formats the number as a currency for the country of the locale.

- $localeNumFormatXdec

  This formats the number with X number of decimals, with X is 0->9-.

- $localeNumFormat

  This variable should always be used in conjunction with the setLocaleNumFormatDecimals(string) function. The function takes the xpath of the current substitution field as an argument, then sets $localeNumFormat to use the correct amount of decimals. This variable is used on numbers without a fixed amount of decimals or values which are not currency amounts.

  Note: Always add the script and the modification when using this variable.

- $localeDateCurrFormat

This formats the date in the standard short format for the locale.

If more alternatives are required, then add the variables to the setGlobalFormats in the global function file.

Example:

$localeNumMediumDateFormat = "date.medium{} (" + $Locale + "){}"; Could be added to use medium date format for given locale.

Another alternative is to use the formats provided in StoryTeller. Only use these formats if specific locale settings are required.

Type attributes

Certain blocks in the XMLIN event has type attributes mapped. To select a specific type of block, xpath can be used.

Example:

Block_Address[Address_Type="Payer"]/Address_Name. To select payer name.

Utility functions

Except for formatting numbers, some utility functions are provided to produce a "zebra rows" effect. Zebra rows are rows in a table with alternating colors. To produce this effect, you can set a table row to the color that you want, and then use the provided functions to set alternating rows to be transparent.

- initZebraRows

  This utility function resets zebra row count. The next call to setZebraRowsHead makes the line transparent. This action must be called before the table where the zebra rows are to be displayed.

- setZebraRowsHead

  This utility function sets the row to a specific color and ensures that the next call to the same function yields a different color. This function must be used on a repeated row.

- setZebraRows

  This utility function sets the current row with the same color as the previous setZebraRowsHead row.

- initZebraRowsHead

  This utility function changes the color for the next call to setZebraRowsHead or setZebraRows. This function works like initZebraRows, but does not need be called on a table row.

  Example:

  For a simple zebra row table, the user can add initZebraRows as a script somewhere before the table, and then add the script setZebraRowsHead to a row under the repeater for.

Languages not supported by the Arial font

The Arial font is used in the layout templates for both the Forms and Base projects. For languages which use characters not supported by the font, a font substitution table is required.

Applying a font substitution table

1. Open the main project InforM3MainProject.dcproject from the location where you installed the XML StoryTeller project template.
2. Double-click GlobalResources resource set.
3. Right-click the table folder. Click New>table. Name the file fontmap.tbl, or something similar.
4. Double-click the created table fontmap.tbl. This action opens the Edit program where you can define the font substitution table. For further details, refer to the StreamServe documentation.

   For example, you can add these lines to change the Arial font to Tahoma:

   ```
   mapfont "Arial"                        "Tahoma"
   mapfont "Arial_Bold"                   "Tahoma_Bold"
   mapfont "Arial_Bold_Italic"            "Tahoma_Bold"
   mapfont "Arial_Italic_Underline"       "Tahoma_Underline"
   ```

5. Save the file and exit.
6. Double-click pltGlobalPlatform.
7. On the Platform screen, right-click the output connector FILE_PDF, then select Settings. On the Output Connector Settings - FILE_PDF screen, select the platform layer. Under Device Driver Settings, click the green square icon to browse to the new font substitution table fontmap.tbl.

   Under Device Driver Settings, click the green square icon to browse to the new font substitution table fontmap.tbl. On the Browse for Resources screen, select the GlobalResources resource set, then select fontmap.tbl. Click OK.
8. Repeat the previous step for other output connectors MAIL_PDF, ATTACHMENT_PDF, and IDM. The steps can also be used for PCL, MATRIX, and MAIL_HTML.
9. Click Save and close the platform screen.
10. To test the font substitution, export and deploy your application. Proceed with normal testing. See "Adding a project to a StreamServe installation" on page 34.

Configuring IDM for M3 standard output

7

IDM is an application used to archive documents and files. For IDM connection, these are the components delivered:

- Standard document types
- Standard business context model for the standard document types
- Standard mapping of output fields to send the document directly to IDM

The configuration data for IDM is found from the downloaded M3_BE_15.1.4_IDM_Configuration_vxx.zip file.

Connecting StreamServe to IDM

To connect StreamServe to IDM, an Application Program Interface or API key to ION is required. This key is generated by a system administrator for an IDM user in Infor Ming.le. The key should be generated for a generic user with the sole purpose of connecting StreamServe to IDM.

After the IONAPI key is generated, the key must be moved to the StreamServe server. Then, configure the IDM connector in OpenText Design Center for the M3 XML StoryTeller base project to point to this key.

Important: The IONAPI key provides login rights to generic users, so it must be stored in a secure location on the StreamServe server.

Importing information into IDM

To enable standard configurations in IDM, you need to import the configuration for the required document types.

The files with Business_Context_Model in the file name are for context mapping and are required for the M3 H5 client. The other files are document types.

These are the steps for importing the document type and business context model into IDM:

1. Extract the configuration data from the downloaded zip file. Select the file you want to import. For the M3 H5 client, this includes both the document type and the business context model.
2. In Infor Ming.le, open Document Management from the app menu.
3. Click the icon with the three horizontal bars next to Document. This opens the Control Center.
4. Click Configuration Exporter, then the Importer tool.

   a. Click Import.
   b. Click Select XML file.
   c. From the file list, select M3_CustomerInvoice.xml. Click Open.
   d. Check the details and click OK.

5. Go back to IDM Control Center by clicking on the Home icon.
6. Click Configuration.
7. Click Document Type filer.
8. Select the imported document type from Document Types in the repository panel. Right-click the document to move Customer Invoice into Document Types from the IDM panel.
9. Click Save.

Limitations

This configuration only supports the M3 H5 client. It does not support IDM running on IBM DB2 Content Manager or MS SharePoint.

Configuring M3 BE for IDM connection

The M3 BE output is delivered with predefined connections to the delivered document types. If you want to make changes to the document types or connections, it can be done in the "XML Structure. Open" program in CMS006. From the F-panel, you can connect a printer file to a document type. From the G-panel, you can configure the document attributes.

Verifying the configuration

In these steps, the Manual Invoice ARS121PF is used as an example. ARS121PF can be replaced with other supported documents for this test. By default, this printout is connected to the document type M3_Customer_Invoice.

1 Start "Output Server. Open" (MNS218). From the program, create a record and define the Streamserve server by specifying the IP address of the server to be used.

2 Start "Output Service Type. Open" (MNS217). From the program, create a record and define the interface as XML.

3 Start "Output Service. Open" (MNS216). From the program, create a record. Specify the created server, the created service type, and the port of the server.

4 Start "Output Service Selection. Open" (MNS204). From the program, create a record. Specify your user name, the printer file as ARS121PF, and the previously created service.

5 Start "Output Media Selection. Open" (MNS205). From the program, create a record. Specify the printer file as ARS121PF and set the media to *ARCHIVE.

In this example, printout ARS121PF is used and it can be reprinted from "Customer Invoice. Display Manual" (ARS122) using Related option 6.

6 Open Infor Document Management to verify if the printout has been added.

7 Start "Output. Manage per Job" (MNS206). From the program, ensure that the output job is saved by checking if the output job has the SAV status.

8 Check the OpenText Control Center log for your StreamServe application to determine if any error occurred.

9 Start "XML Structure. Open" (ARS205). Select an invoice from the list. Once the invoice is selected, it will appear in the Related Information pane.

Optional: Version control for documents

Version control for documents is an alternative way of uploading documents to IDM. The standard behavior is to always upload new documents from StreamServe to IDM. With version control, the documents are updated instead, and the document's version history is saved. The method of updating documents is defined in IDM connector in the OpenText Design Center.

1 To enable version control for documents, go to the OpenText Design Center. Go into the runtime for the document you want to work on, for example, rtInvoices.

2 In the runtime, open the IDM connector.

3 To enable version control, set the value of "Document version control" to true.

4 To define how to find a document if an existing version exists in IDM, set the value of "Use attribute value for document exist check" to true for all attributes that must be used for the exist check.

Note: The order of the "Use attribute value for document exist check" check box must match the order of the attribute IDs defined in the M3 program (CMS006). Attributes for company, division, and accounting entity should always match for the exist check to succeed.

5 After these settings, perform an export and redeployment of the project. When an IDM upload is performed, the document exist check is done. Depending on the result the of this check, these are the effects:

   a If a matching document is found and it is updated, you can view its history in IDM. All attributes not used in the exist check will be updated.

   b If no matching document is found, a new document is added.

   c If more than one document is found, there will be an error in the connector and the process will be stopped.

Optional: Import configuration data for Russia documents

8

Infor delivers configuration data for the Russia-specific documents. Each set of configuration data is delivered as an M3 BE package, including XML files in zip format. These packages are made available at Infor Product Download Center, and must be deployed through the LifeCycle Manager or LCM client.

Note: The naming convention for M3 BE packages containing configuration data is M3_BE_15.1.4_ConfigData_RussiaDocuments_vxx.zip.

Preparing the import of configuration data

LCM reads the XML file and uses standard API programs in M3 BE to import the templates. During import, you can decide whether existing configuration data for templates is kept or not. If you choose to replace existing records, Delete API transactions is run before the add API transactions.

1. Identify the location of the package at Infor Product Download Center.
2. Download the package or packages to a folder at the LifeCycle Manager Server.
3. In the LCM client, navigate to the host tab and expand the host where M3 Business Engine is installed.

Uploading additional M3 BE packages

1. Right-click M3BE_15.1, and select Upload M3 BE packages.
2. Click Select. Select the directory on the administrative client that contains the M3 Business Engine package. You can select several packages at a time.

   Tip: To select all packages, right-click the list and click Select all. Select your preferred packages, then Click Next.

3 On the Summary screen, verify the property values, then click Finish.

4 When the task is finished, a dialog box appears. Click OK. To view the log file, either click View log or select the Logs view option.

Importing configuration data

Use this procedure to import database configuration data. For example, to import configuration data from a zip file to the database, these are the steps:

1 Right-click your target environment and select Database>Import Configuration Data.

Note: You can also click Import Configuration Data under Database on the dashboard.

2 These are the next steps based on the environment status:

- If the environment is running, you will be requested to authenticate. Proceed with the next step.
- If the environment is stopped or is in maintenance mode, you will be requested to start the environment. Select Start Environment. Click Next.

3 Specify an M3 BE user name and password. Click Next.

Important: Ensure that the M3 user is connected to the system configuration for Russia (MRU).

4 From the Select Configuration Data Package field, specify this information:

Select packages to show Select whether you want to list all available packages or hide the packages which were already applied.

Package Select the zip file containing the configuration data. Zip files listed here are stored either in the **/M3BE/_db_data/config_data** folder, or on the LifeCycle Manager server.

Replace existing records The records in target that have a duplicate in the source are replaced by the imported ones.

Keep existing records The existing records in both source and target are left unchanged in the target.

Company/Division Optionally, specify whether or not the configuration data should be imported for a specific company and division.

Click Next.

5 On the Summary tab, verify the values provided. Click Finish.

6 When the task is finished, a window is displayed. Click OK or click View log. You can also click the Logs tab to view the log file.

The database configuration data is now imported.