TIBCO Foresight® Products

Using XML

August 2017

Two-Second Advantage®

Important Information

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

TIBCO, Two-Second Advantage, TIBCO Foresight EDISIM, TIBCO Foresight HIPAA Validator Desktop, and TIBCO Foresight Instream are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Enterprise Java Beans (EJB), Java Platform Enterprise Edition (Java EE), Java 2 Platform Enterprise Edition (J2EE), and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle Corporation in the U.S. and other countries.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

THIS SOFTWARE MAY BE AVAILABLE ON MULTIPLE OPERATING SYSTEMS. HOWEVER, NOT ALL OPERATING SYSTEM PLATFORMS FOR A SPECIFIC SOFTWARE VERSION ARE RELEASED AT THE SAME TIME. SEE THE README FILE FOR THE AVAILABILITY OF THIS SOFTWARE VERSION ON A SPECIFIC OPERATING SYSTEM PLATFORM.

THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

General Contact Information
TIBCO Software Inc.
3303 Hillview Avenue
Palo Alto, CA 94304 USA
Tel: +1 650 846 1000
Fax: +1 650 846 1005

Technical Support
E-mail: support@tibco.com
Web: https://support.tibco.com
(Note: Entry to this site requires a username and password. If you do not have one, you can request one. You must have a valid maintenance or support contract to use this site.)

Contents

Introduction

Intended Audience

This document is intended for users who wish to use TIBCO Foresight programs with XML data.

You will need a basic knowledge of XML and of the TIBCO Foresight programs that you will use with XML as described in the next section.

TIBCO Foresight Products that work with XML

TIBCO Foresight products can use XML on Windows platforms only.

Program	What you can do	For more information, see …
TIBCO Foresight® EDISIM® 6.7 or later (Standards Editor, Doc Builder, EDISIM® Validator)	Import, create, edit, print schemas and DTDs Validate XML data Create documentation	EDISIM's Documentation directory: - **TIB_fsp_edisim_<*n.n*>_fseditor.pdf** - **TIB_fsp_edisim_<*n.n*>_fsdocbld.pdf** - **TIB_fsp_edisim_<*n.n*>_fsvalidator.pdf**
TIBCO Foresight® HIPAA Validator® Desktop	Validate XML data	See **TIB_fsp_edisim_<*n.n*>_fsvalidator.pdf** in HIPAA Validator® Desktop's Doc directory
TIBCO Foresight® Instream® (validate, Document Splitter)	Validate XML data Split into good and bad data	Instream®'s Doc directory: - **TIB_fsp-instream_<*n.n*>_usersguide .pdf** - **TIB_fsp-instream_<*n.n*>_docsplitter.pdf**

Overview

You can:

- Import a schema or DTD into EDISIM Standards Editor, where you can edit it, including adding business rules.
- Validate XML data against the schema or DTD in EDISIM Validator, HIPAA Validator Desktop, or Instream.
- Split good from bad data with Document Splitter.

Since XML is not firmly based on standards the way EDI is, you will notice differences in how it is implemented in TIBCO Foresight products:

- EDI has guidelines, XML has schemas and DTDs.
- You must have EDISIM to use XML at TIBCO Foresight. It creates a TIBCO Foresight "guideline" (STD file) from the schema or DTD.
- EDISIM Standards Editor imports, edits, and exports schemas and DTDs. Any valid schema or DTD will work.
- In EDISIM, XML enumerated values are stored in application value lists rather than code lists.
- You can create a schema or DTD from scratch in EDISIM but it is currently much easier to create it another way and then import it.
- APF files are honored by XML validations.
- Instream validates and splits XML under Windows only.
- HIPAA Validator® Desktop's Library does not work with XML guidelines.
- Some XML resources
    - http://www.w3schools.com/xml/default.asp
      XML tutorial
    - http://www.w3schools.com/schema/default.asp
      Schema tutorial
    - http://www.w3.org/XML/Schema
      Schema reference
    - http://www.xml.com/axml/testaxml.htm .
      Annotated XML specification

Demos

For a complete list of demos, see **Demo_Index.pdf**.

Program	Data file	Validate with …
EDISIM	**XML_PO_f.xml** **XML_PO_g.xml** in EDISIM's Samples directory	**XML_PO_F** **XML_PO_G** (use EDISIM Validator)
HIPAA Validator Desktop	**XML_PO_f.xml** in HIPAA Validator Desktop's DemoData directory	**XML_PO_F**
HIPAA Validator Desktop	**XML_275.txt** in HIPAA Validator Desktop's DemoData directory	Start with **275_X151** Select **XHL7** when it reaches the Bin segment
Program	**Script**	**Output**
Instream	**V_XML_PO** in Instream's Scripts directory	**XML_PO_f_Results.txt** or **XML_PO_Results.txt** in Instream's Output directory
Instream (HIPAA)	V_DS_XML_split_PO ValidationHighlighter_XML	Various files in Instream's Output directory
Validation Highlighter	**ValidationHighlighter_XML** in Instream's Scripts directory	**XML_PO_f.html** or **XML_PO.html** in Instream's Output directory
Translator	T_837I_4010_to_XML_and_back	**837Iclean_edi.txt** **837Iclean_xml.xml** In Translator's Output directory

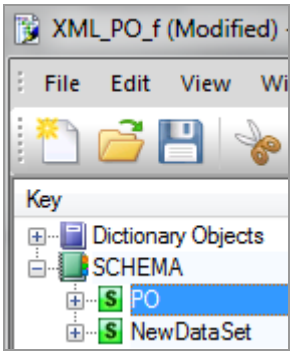
Tutorial

This tutorial will walk you through the most basic steps in editing a schema and validating XML data with it.

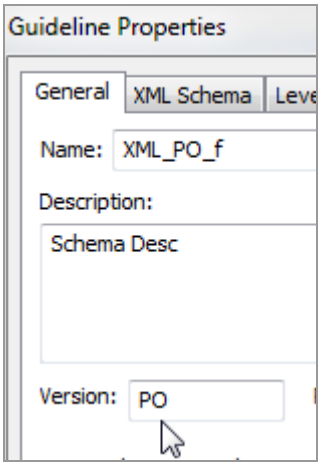
Importing the Schema into EDISIM

1. Open EDISIM Standards Editor.
2. Choose **File | Import | Import XML Schema and open**.
3. Go to EDISIM's **Samples** directory and choose **XML_PO_f.xsd**.
4. In the Import Standard box, use **XML_PO_f** and click **OK**.
5. Set up the version.

   To do this, expand the SCHEMA line and notice that the root element is called PO. In TIBCO Foresight XML guidelines, we use the name of the root element as the version.



   Choose **File | Properties** | **General** tab and type `PO` for Version (it will initially have VRI, which is just a placeholder):

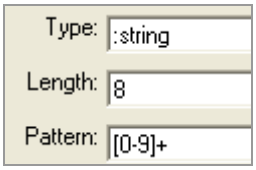


Editing the Schema in EDISIM

1. Expand the PO element and select the **PO_NUMBER** element:

2. In the bottom pane, enter `Purchase Order Number` for **Description** and change its **Minimum Occurrences** to **1** to indicate that it must be used.

3. Select **PO_DATE** in the top pane. In the bottom pane, enter `Purchase Order Date` for its Description, and add this **Length** and **Pattern**:

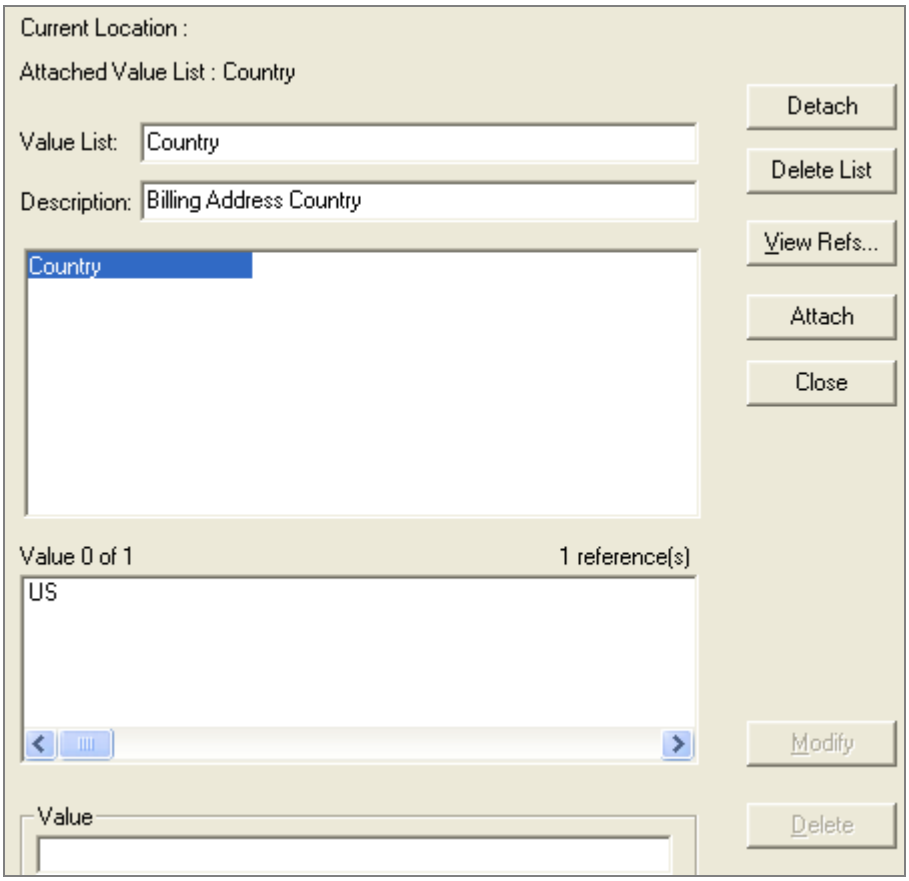


   The pattern can be anything from the full set of regular expressions. This one means that it can contain only digits. See the Help menu for more examples.

4. Acceptable XML data values are called enumerations. These go in application values lists rather than in the code value pane.

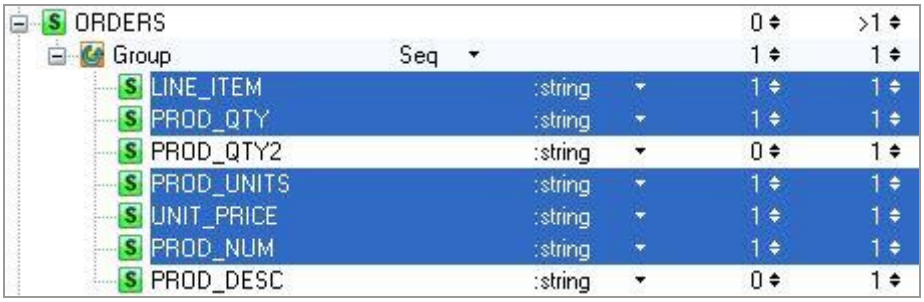
   Right-click on **BT_COUNTRY** and choose **Application Values…** .

Set up the application value list like this:



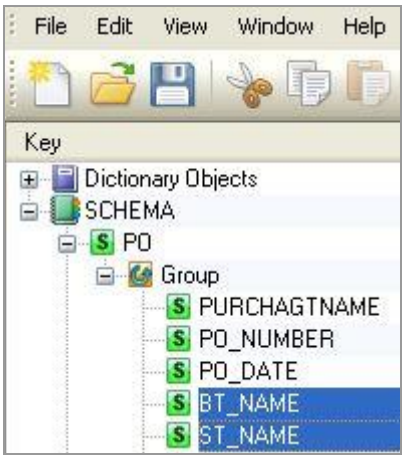
Click **Attach**. This requires that the value for this element always be `US`.

5. Under **ORDERS | Group**, set the Min to **1** in the top pane for
   LINE_ITEM
   PROD_QTY
   PROD_UNITS
   UNIT_PRICE
   PROD_NUM



This means that these are required.

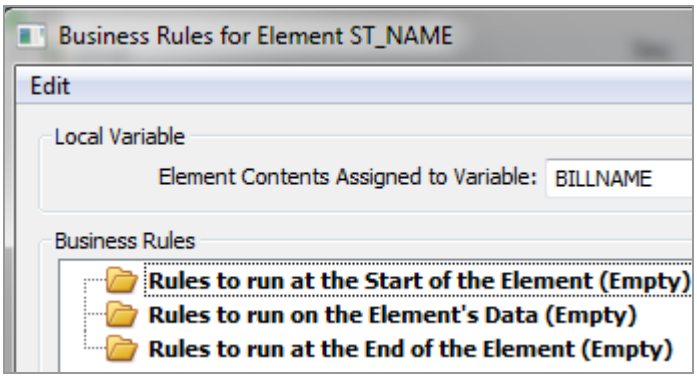
6. We want to ensure that we get either a BT_NAME or a ST_NAME.



Set up a variable on the first one.

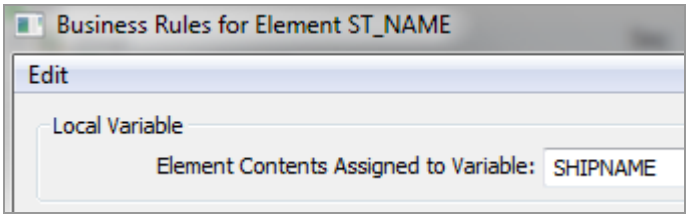
Right-click on **BT_NAME.**

Capture the contents in a variable called BILLNAME:



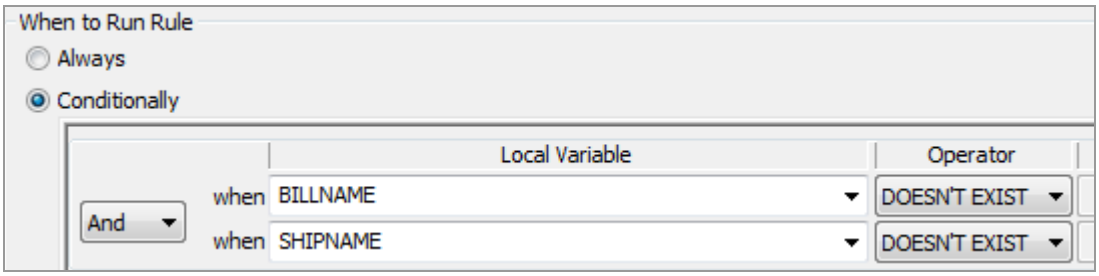
Click **OK**.

7. Right-click on the other element in the condition, **ST_NAME**, and set up this variable:

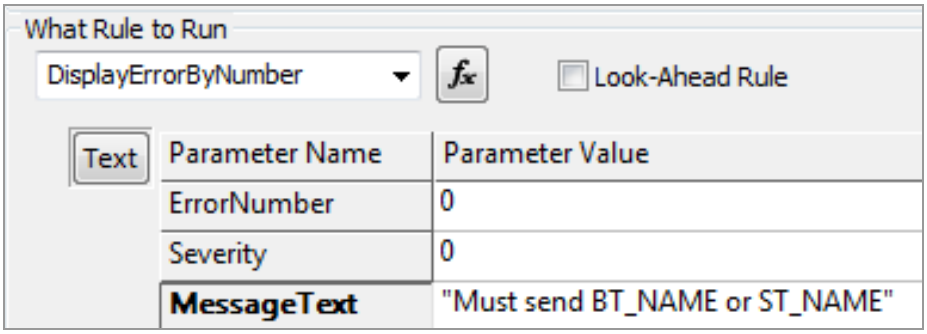


Close and re-open the business rules dialog and click NEW:

Make these selections in the **When to Run Rule** area:



Select **DisplayErrorByNumber** in the What Rule to Run drop box, and set this up to display an error message if the elements don't exist. (If the grid doesn't appear, click the Text button.)



Click **OK** twice.

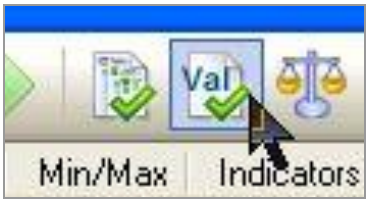
8. Save.

See **BusinessRules.pdf** for more information.

Validating data with EDISIM Validator or HIPAA Validator Desktop

You can now validate XML data using **XML_PO_f**.

1. Click the EDISIM Validator toolbar button within Standards Editor:



This opens EDISIM Validator, which is installed in EDISIM's Bin directory.

(If you want to use HIPAA Validator Desktop instead, copy XML_PO_f.std from EDISIM's User Files\Public Guidelines directory to HIPAA Validator Desktop's Database directory and then open Desktop. You may have the HIPAA Validator Desktop toolbar button in Standards Editor.)

2. Choose **File | Open**, navigate to EDISIM's **Samples** directory, and select **XML_PO_f.xml**.

3. In the Select Standard box, choose our XML guideline **XML_PO_f**.
4. When the validation completes, click on the blue errors at the top and look at the XML data in the bottom pane to see where the error occurred:

   USA is not valid. The application value list specified a value of US.

   PROD_NUM appears twice and it is only allowed once.

Validating data with Instream

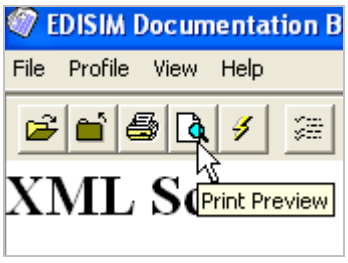
Guideline XML_PO_f is already installed in Instream's Database directory.

1. If you would like to use the guideline that you created, move the one that was installed is Instream's Database directory (renaming the STD file won't do) and copy yours there from EDISIM's **User Files\Public Guidelines**.
2. Go to Instream's **Scripts** directory and view the contents of the XML_PO_F script. Notice that the guideline parameter is **-xgXML_PO_F**. The **–xg** indicates an XML guideline.
3. Close the script and execute it.
4. Go to Instream's **Output** directory and view the contents of **XML_PO_f_Results.txt**. Notice the error detailed in the **EDTL** record and the following EMSG record. These records are described in **TIB_fsp-instream_<n.n>_usersguide.pdf** in Instream's Doc directory.

Printing Guideline Documentation

To create formatted documentation:

1. Open EDISIM Document Builder.
2. Choose **File | Open Standard**, choose **XML_PO_f**, and click **OK**.
3. Click **Print Preview**:



4. Click in the document to see the various levels of magnification and use the scroll bar to see the entire page.
5. Use the **Next Page** button to see other pages.

6. Click **Close** to close Print Preview.
7. Close Document Builder.

See Printing DTDs and schemas on page 37.

Using Standards Editor with Schemas and DTDs

Importing a Schema or DTD

To import:

1. Open Standards Editor.
2. To import a schema, choose **File | Import | Import XML Schema and open**. To import a DTD, choose **File | Import | Import XML DTD and open**.
3. Navigate to the schema or DTD file and then choose **Open**.

During the import, Standards Editor saves it as a STD file in EDISIM's **User Files\Public Guidelines** directory.

If the schema has import or include elements, the referenced schemas will be imported as part of the master schema that contains them – if they are available to EDISIM.

You can then edit it or:

- Use EDISIM Validator to validate XML data.
- Use HIPAA Validator Desktop or Instream to validate XML data.
- Use Docsplitter to split an XML file into good and bad XML data files.

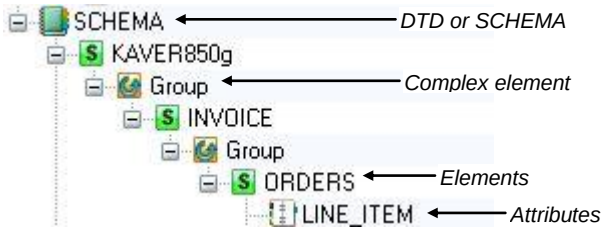
Opening a Schema or DTD

If you have already imported your schema or DTD into EDISIM, you can open it just as you do any other guideline:

1. Open Standards Editor.
2. Choose **File | Open** and choose the **User Guidelines** tab.
3. Select the guideline and click **Open**.

Top Pane

The Standards Editor top pane for an XML guideline looks similar to the top pane for an EDI guideline in that it has a Dictionary Section and a local SCHEMA or DTD section (like a Set or Message).

Column	Contents
Key	The nested list of objects in the SCHEMA or DTD: 
Group	Available on group elements to define the order and requirement of elements it contains. SCHEMA or DTD order indicators that define the order in which the child elements can occur: **Group** – Use only in the dictionary when defining a complex type. Defines this as a group of elements. **All** - Child elements can appear in any order; each child element can occur 0 or 1 time. **Choice** – Only one of the child elements can occur. **Seq** - Child elements must appear in the sequence in which they are listed.

Type	The element's type. If it is a built-in XML type, it is preceded with a colon. Examples:
	`:date`
	`:integer`
	If it is from one of the Types dictionaries, it does not have the colon.
	To change the type, click on it in the top pane, or use the bottom detail pane's **Type** field or **Change Type** button.

Use	*This column is not used for XML guidelines*
Min	The minOccurs occurrence indicator: the element must occur at least this many times at this location in the data 0 means the element is optional and can be left out of the data altogether. Any other number means it must be used at least that many times.
Max	The maxOccurs occurrence indicator: the element cannot appear any more than this number of times at this location in the data. >1 means unbounded
Indicators	These abbreviations are visual cues to characteristics such as level notes, business rules, code values, and other items that do not always appear on the screen. For a list, search for "indicators" in **TIB_fsp_edisim_<*n.n*>_fseditor.pdf**.

Bottom Pane

Details View Pane

The Details View pane can be floating or docked. By default, the pane appears docked at the bottom of the Standards Editor window. Single click the top bar of the pane and drag the pane to a new position. Double click the top bar of the pane to toggle between its most recent floating and docked positions. Size and position are saved when you exit the guideline.

Not all fields appear for every object in the top pane.

Field	Contents
Description	For your use.
ID	For your use.
Reference	This element points to another item for its definition. To identify the item to which it points, click the **Change Type** button at the end of the Type field, choose **Reference**, and use the drop-down list.
Type	Same as Type column in top pane. You can type in the type, or use the Change Type button at the end of the line. See Types on page 18.
Length	For data that must be a fixed length. An integer that specifies the exact number of digits or characters for the value.
Min Length	An integer that specifies the minimum length for the value.
Max Length	An integer greater than 0 that specifies the maximum length for the value.
Pattern	An optional regular expression to describe the sequence of characters that are acceptable in the value. Use this instead of the application value list regular expressions unless you need to use an application value list to enforce literal values or multiple patterns.
Abstract	This element is abstract, meaning it cannot be used in the data. Instead, should be replaced by a member of its substitution group. If this is a type, declaring it abstract requires the use of another type that is derived from it.
Nillable	The element must be present in the data but can be empty.
Substitution Group	The current element may be substituted for the element entered here.
Form	Optional. Determines whether this item has to be qualified with the namespace prefix. Default is the value of the elementFormDefault attribute of the schema element. Not available for elements directly under the schema element.
Block	Determines whether other elements can be substituted for this one.

Field	Contents
Final	Only used on elements that have the schema element as a parent. Sets the default value of the final attribute.
Minimum Occurrences	Same as Min column in top pane.
Maximum Occurrences	Same as Max column in top pane.
Default Value	For optional elements, this value will be assumed if the element appears in the data without a value. If the element is not in the data at all, then the default value is ignored. Does not apply for required elements, since the XML must contain a value. For attributes, this value applies even when the attribute is missing in the data.
Fixed Value	If the XML data includes this element or attribute, it must contain this value. If the XML omits this element or attribute, this value will be assumed.
Edit Notes	Add level notes to this item. These are for your own use and can optionally print in Doc Builder.
Edit Rules	Add business rules to this item. Please see **BusinessRules.pdf**.

Code Values Pane

The Code Values pane is not used for XML. The XML equivalent, enumerations, should be put into an application values list and attached to the element.

Cross Reference Pane

Cross-referencing is not available with XML data.

Editing a Schema or DTD

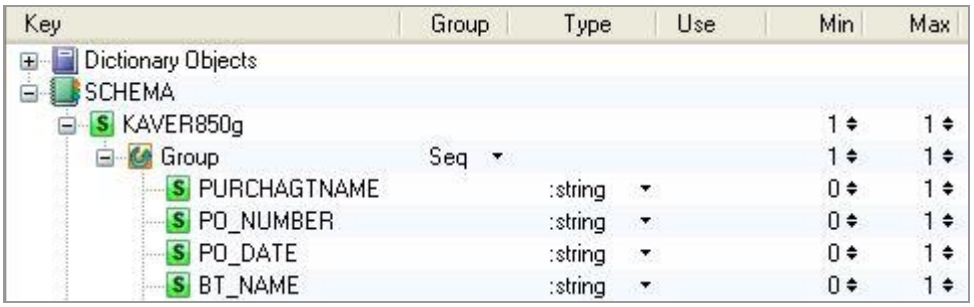
After importing, you can edit it much like you would any EDI guideline.

SCHEMA Section

The local SCHEMA or DTD section contains the global elements.

Each object has a Type field that identifies that object's base type, if it has one. The basic w3c schema types all start with a colon: `:string, :TOKEN,` etc. You can define others under Dictionary Objects.

In the top pane, open the root element SCHEMA to see the elements and columns that describe them.



Values

Enumerations define a list of acceptable values. The Code Values tab is not used with XML data since the data does not necessarily conform to EDI limitations for codes.

Instead, use any of these with simple elements or attributes:

- Application value lists to enforce specific values
- The Pattern field in the bottom pane to enforce a pattern
- Business rules like CodeLookup|FindUserCode

Application Value Lists

This is one way to enforce literal values and patterns (regular expressions). This operates exactly like application values when using EDI data. Please see Application Value Lists in **TIB_fsp_edisim_<*n.n*>_fseditor.pdf**. They are best for values that do not change much.

Pattern field

The bottom Detail pane contains a Pattern field where you can type in any regular expression to enforce a pattern. For example, to require that the value start with TC and be followed by digits, you would type this:

```
^TC[0-9]+
```

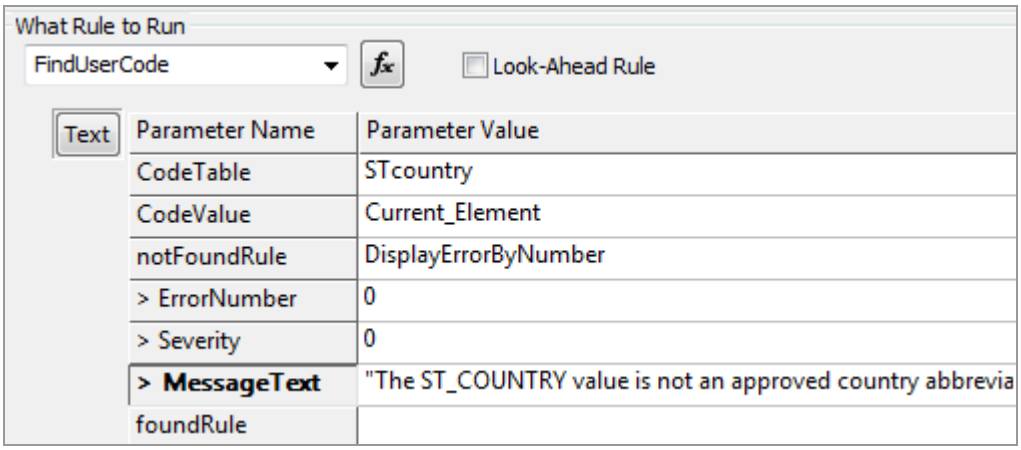
Please see Regular Expressions in **TIB_fsp_edisim_ <n.n>_fseditor.pdf** for examples and details.

Business rules to enforce values

Many extended business rules enforce values: List|ListCheck rules, CodeLookup |FindUserCode, etc.

Example

This rule checks an external list of values in a separate file, and returns an error message if the current value is not in the list:



The list called **Stcountry** would be in a file designated by the UserTable entry in EDISIM's or Instream's $Dir.ini. Please see **BusinessRules.pdf** for details.

Types

Creating a Type

The Dictionary lets you create types in:

- The Simple Types Dictionary, which contains Simple Types and Global Attributes, and
- The Complex Types Dictionary, which contains Complex Types and Attribute Groups

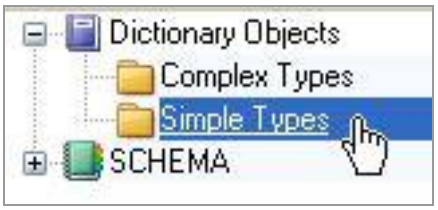
You can create new types in the dictionary and then use them in the SCHEMA. For elements that are used more than once, this is a good alternative to simply adding them directly in the SCHEMA.

It also lets you edit the type in one place (the dictionary) and have it affect all places where it is used as a type in the SCHEMA.

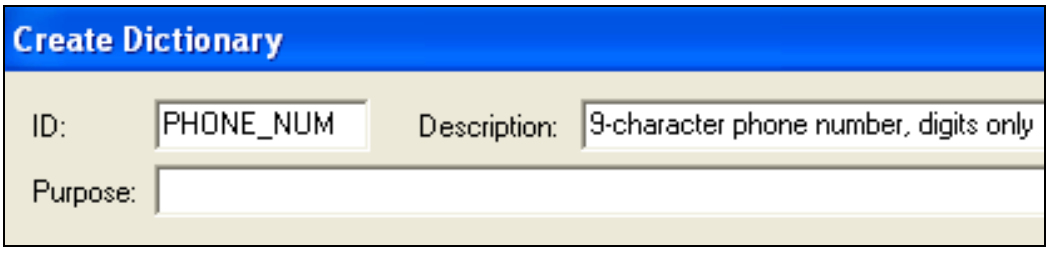
Example - Simple Type

Create a simple type PHONE_NUM in the dictionary and then use it twice in the schema.

1. Open **Dictionary Objects** at the top and click **Simple Types**.

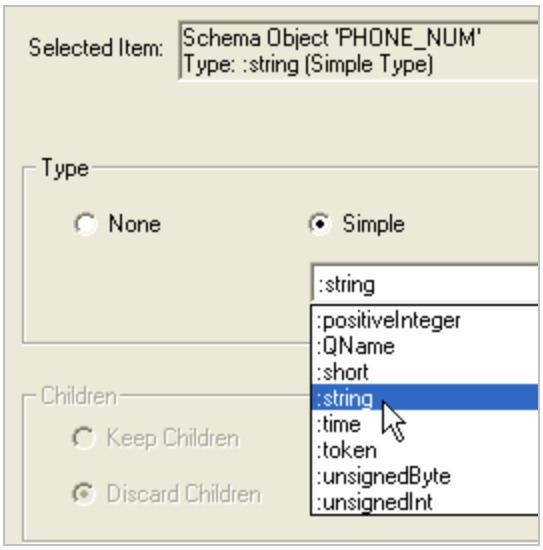


2. Select **Edit | Create new item … | Schema Simple Type** and fill out the box like this:



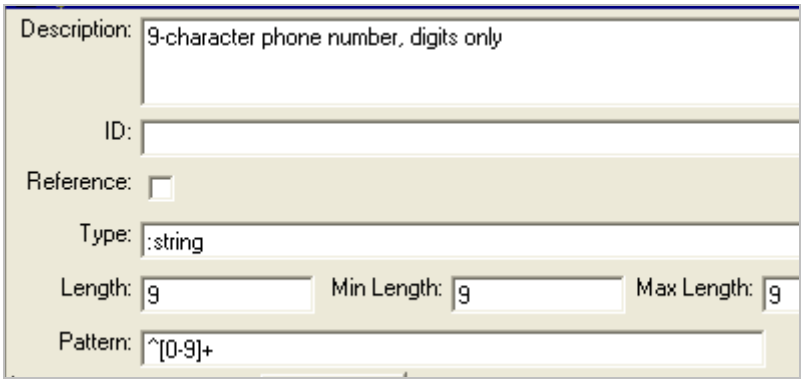
Close the box.

3. Click in the Type column for PHONE_NUM and select **:string** from the drop-down list.



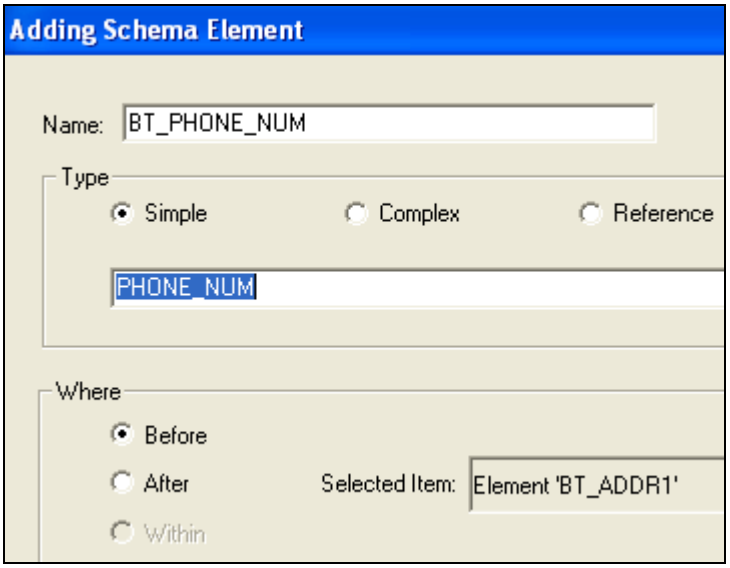
4. Click on another line in the top pane, and then click again on PHONE_NUM to refresh the bottom pane.

5. Fill out the bottom pane like this:



6. Return to the SCHEMA and click BT_ADDR1:

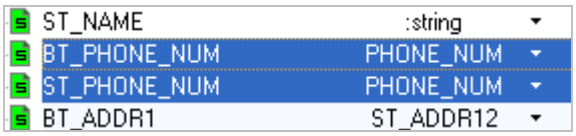
7. Choose **Edit | Add item | Schema Element** and fill out the box like this. PHONE_NUM will be at the bottom of the drop-down list after the built-in types.



There is no colon at the beginning of a type that you create.

Finish adding the element.

8. Add ST_PHONE_NUM after BT_PHONE_NUM.



Example - Complex Type

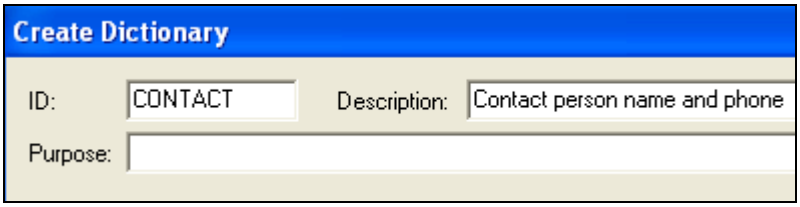
Create a complex type CONTACT in the dictionary and then use it in the schema.

1. Under Dictionary Objects, click **Simple Types** and create this type in the Simple Types dictionary:

   `NAME  Type :string  Min 3  Max 20`

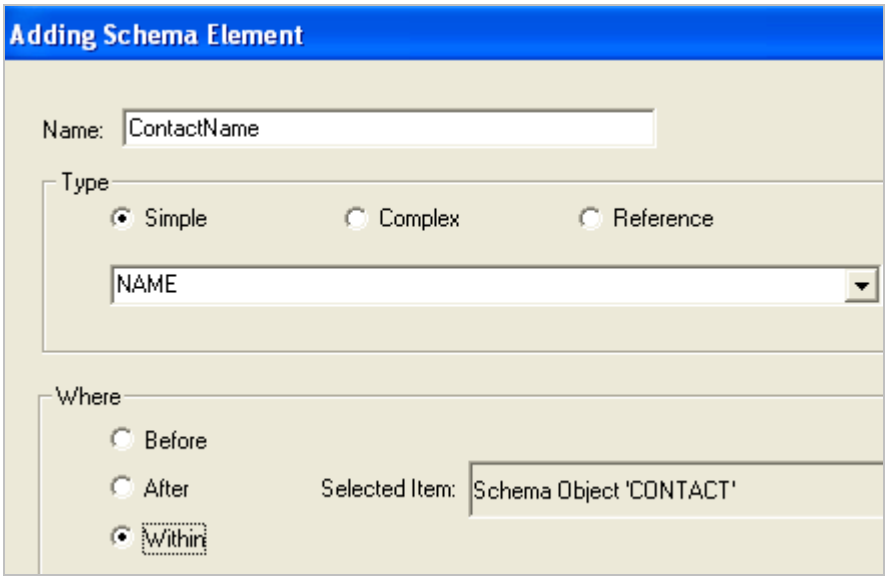
2. Under Dictionary Objects, click **Complex Types**.

3. Select **Edit | Create new item … | Schema Complex Type** and fill out the box like this:

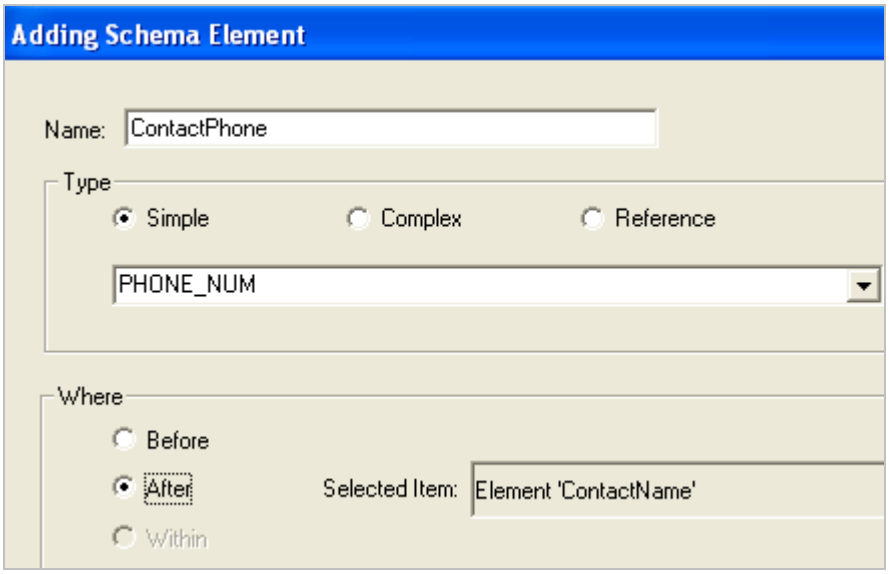


Close the box.

4. Save.
5. With CONTACT selected, choose **Edit | Add item … | Schema Element** and add this simple element:



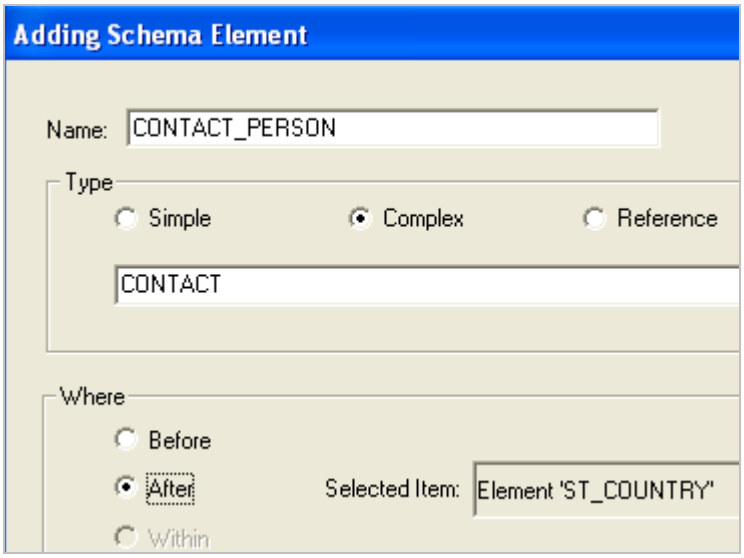
6. Highlight ContactName and add this after:



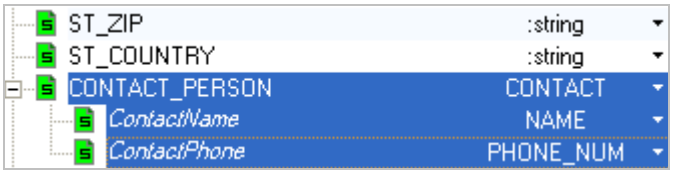
7. Save.

Now use this type in the SCHEMA:

1. Click **ST_COUNTRY** in the SCHEMA and choose **Edit | Add item | Schema Element**.
2. Fill out the box like this:



3. Save.
4. Expand CONTACT_PERSON in the SCHEMA to see this:



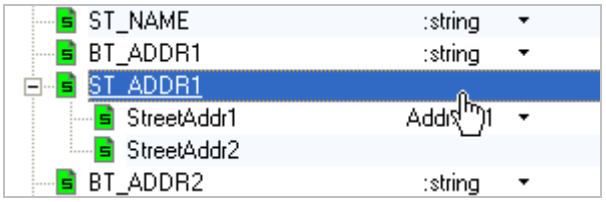
Notice the Italic ContactName and ContactPhone, indicating that these are being supplied from the Complex Types dictionary.

You can use this type as often as you'd like. It is the same as having a local complex group that contains NAME and PHONE, only you can maintain it centrally from the dictionary.

Changing an Element's Type

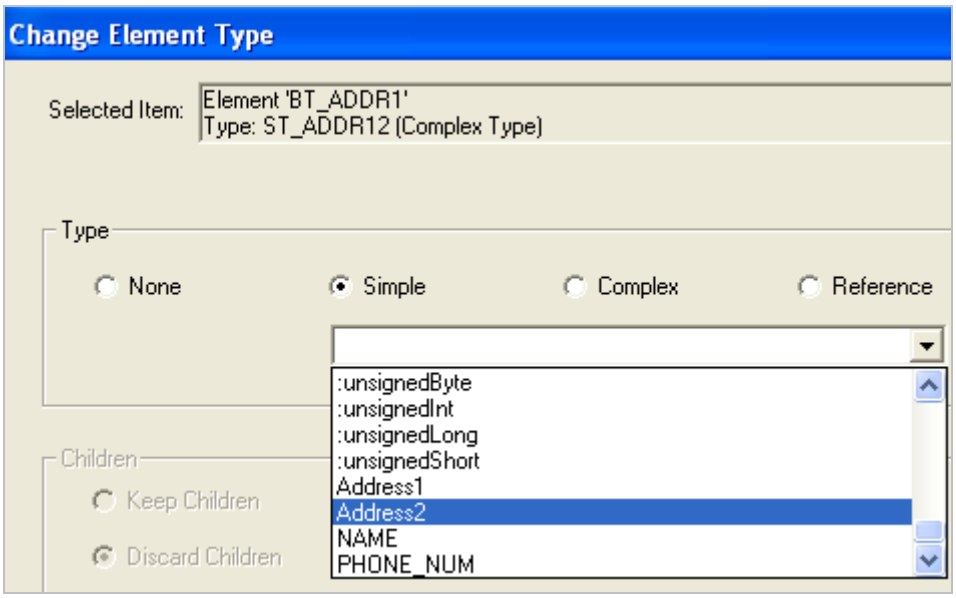
You can change an element's type by:

- Clicking (or double-clicking, depending on your Preferences) on its type in the top pane:



- Or, by clicking the **Change type** button at the end of the Type field in the Detail pane:

This displays the Change Element Type dialog box:



Where:

Selected Item Name and type of the current element. Parentheses at the end of the type line contain one of these:

- **Complex** Selected type has child elements.
- **ComplexContent** Selected type contains data and has child elements.
- **Simple Type** Selected type contains data.

Type Select a type:

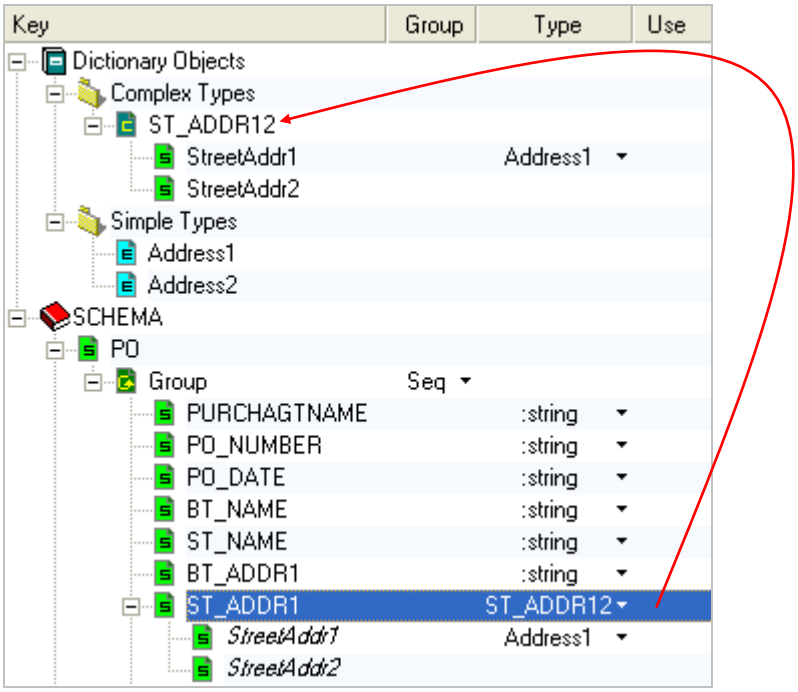
- **None** Deletes the type from the element.
- **Simple** Drop-down list offers all built-in simple types or those from your Simple Types dictionary.
- **Complex** Drop-down list offers all complex types from your Complex Types dictionary.
- **Reference** Drop-down list offers all types defined elsewhere in the document.

Keep children Save the current child elements and add the one(s) from the selected type. Available if you have child elements that are local (see page 25).

Discard children Delete the current child elements and add the one(s) from the selected type. Available if you have child elements that are local (see page 25).

Localizing an Element

If an element uses a complex type from the dictionary, its children will display in Italic in the top pane. In this example, the ST_ADDR1 complex element in the SCHEMA has a type of ST_ADDR12 – which you can see in the Complex Types dictionary. STR_ADDR1 is simply pointing to the dictionary element for its subordinate structure.

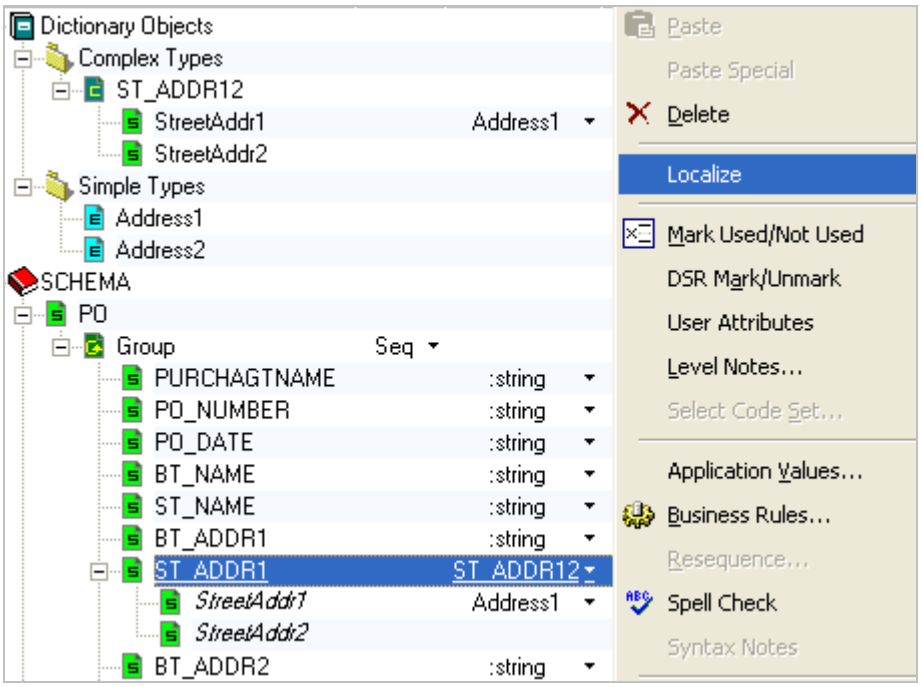


To remind you that it referring to a dictionary type, Standards Editor displays the subordinates *StreetAddr1* and *StreetAddr2* in an Italic font.

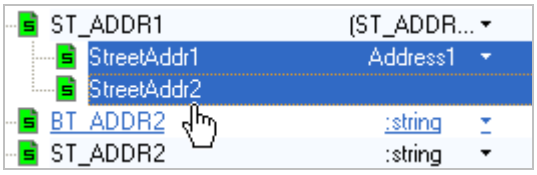
You cannot change anything in ST_ADDR1 in the schema. Changes must be made in the dictionary, and they then will be reflected in all schema places where that element type is used.

If you want to make changes that are local to just one place in the schema, you can "localize" the complex element:

1. Right-click on the complex element that is using the dictionary type.
2. Choose **Localize**.



The subordinates are no longer in Italic and the type is in parentheses – indicating that it was based on the type shown, but is no longer attached to it.

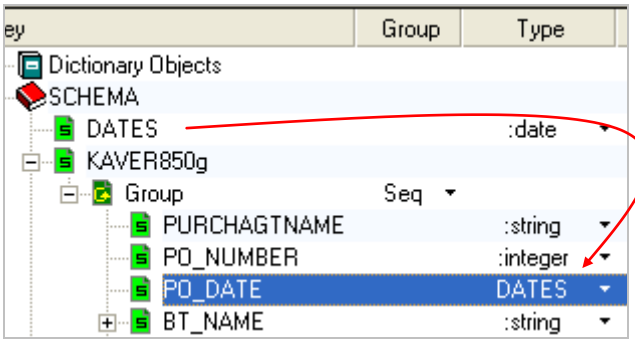


You can now edit the element in the SCHEMA. It will not affect any other location. If there are other dictionary types within this element, they remain attached to their complex types unless you localize them.

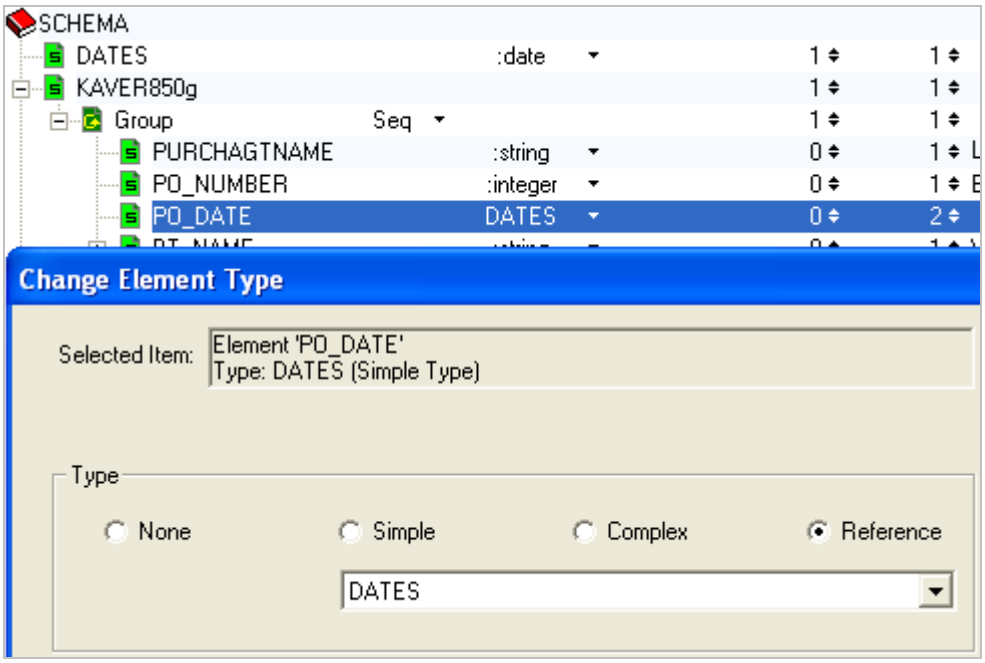
To re-attach a localized element to its dictionary element, right-click on it and choose Localize.

Reference Example

In this example, PO_DATE under KAVER850g has Reference selected in the detail pane so that it uses the definition of DATES under SCHEMA:



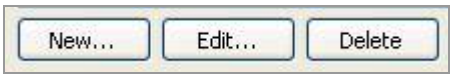
PO_DATE's Type is set up like this:



Business Rules

In the business rules dialog box, you can define variables with XML:

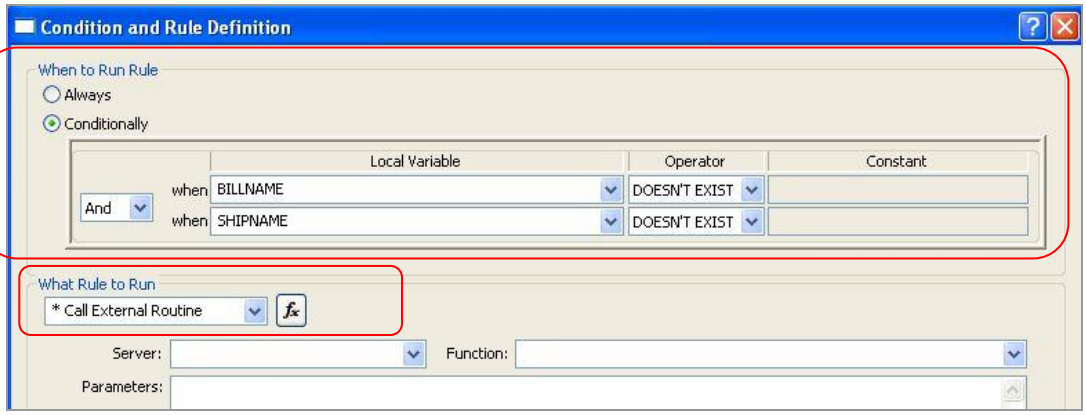
You can also use all of these buttons with XML:



With XML, you can use these features of the business rules dialog box:

You can use all parts of the Variable, Operator, and Constant area with XML

*You can use any rule (such as AddVar or CompareDate) or any function under * Call External Routine with XML*



External routines you can use with XML	
Rule	**Use with XML?**
Reserved variables	
Current_Date	✓
Current_Delim	
Current_Element	✓
Current_ ErrCount	✓
Current_LoopCount	
Current_LoopKey	
Current_Row and Next_Row	✓
Current_Time	✓
GLOBAL_FILENAME	
GLOBAL_FILEPATHNAME	
Using reserved variables in a message	✓
BusinessRules.CCI	
CCIInit	
CCICollect	

External routines you can use with XML	
Rule	**Use with XML?**
CCIAnalyze	
BusinessRules.CodeLookup	
FindCode	✓
FindCodeWithDate	✓
FindUserCode	✓
FindUserCodeWithDate	✓
ValidateZipState	✓
BusinessRules.CustRec	
DefineCustomRec	✓
OutputCustomRec	✓
RemoveCustomRecord	✓
BusinessRules.DateTime	
CheckDateInRange	✓
CompareDate	✓
DateCalc	✓
GetGMTDateTime	✓
ValidateDateTimeUN and ValidateDateTimeX12	
BusinessRules.DBServer	
DBExecute	✓
DBQuery	✓
InvokeWebService	✓
BusinessRules.Exits	
ClearExits	
KeepOrder	
SetCompositePreExit	
SetElementPostExit	
SetLoopPostExit	
SetLoopPostInstanceExit	
SetSegmentPreExit	
UserExitWithoutWait	
UserExitWithWait	

External routines you can use with XML	
Rule	**Use with XML?**
BusinessRules.List	
ClearList	✓
InList	✓
ListCheck	✓
ListContig	✓
ListCount	✓
ListGetVar	✓
ListInsert	✓
ListMinMax	✓
BusinessRules.Looping	
ForEach	✓
Next	✓
ExitLoop	✓
BusinessRules.ODBC	
Setting up your ODBC Connection String	✓
DBOpen	✓
DBClose	✓
DBQuery	✓
DBExecute	✓
BusinessRules.Run	
RunAlways	
RunNoData	
BusinessRules.Substitute	
DeleteSegment	
InsertSegment	
MakeKey	
Substitute	
SubstituteFind	
SubstituteReplace	

External routines you can use with XML	
Rule	**Use with XML?**
BusinessRules.Utilities	
AppendString	✓
BuildString	✓
ChangeCase	✓
ChangeElmAttribute	
CheckFormat	✓
CreateFSUID	✓
DisplayErrorByNumber	✓
FindString	✓
GetToken	✓
Identify	✓
IdentifierLookup	
InsertIdentifier	
Match	✓
MatchApplList	
Normalize	✓
Numbers	✓
OracleLookup and OracleLookupWithDate	✓
OutputCTX	
ReplaceChars	✓
ReplaceString	✓
SetCheckCTT and SetCheckCTTCount	
SetIdentifier	
SubString	✓
Trim	✓
TrimWhitespace	✓
BusinessRules.Variable	
SetLocalVariable	
SetVar	✓
AddVar	✓
Divide	✓

External routines you can use with XML	
Rule	**Use with XML?**
DumpVars	✓
Balance	✓
CompareNstring	✓
CompareNumeric	✓
CompareString and CompareStringNoCase	✓
Clear	✓
ClearLocalVariable	✓
FileTable Rules	✓
GetInfo	✓
GetLength	✓
GetValueFromSegment	
IsAlpha	✓
IsAlphaNum	✓
IsNum	✓
SaveCurrentSegment	
CheckCTT	
CheckDigit	
X12 234-235 CheckDigit	
EDIFACT 3039-3055 CheckDigit	
Other CheckDigit options	
User Defined check digit	
DateTime	✓
FSVBExit.DisplayMessage	
ProductUtilities	

Business Rules that Do Not Apply to XML

Additionally, these do not work with XML files:

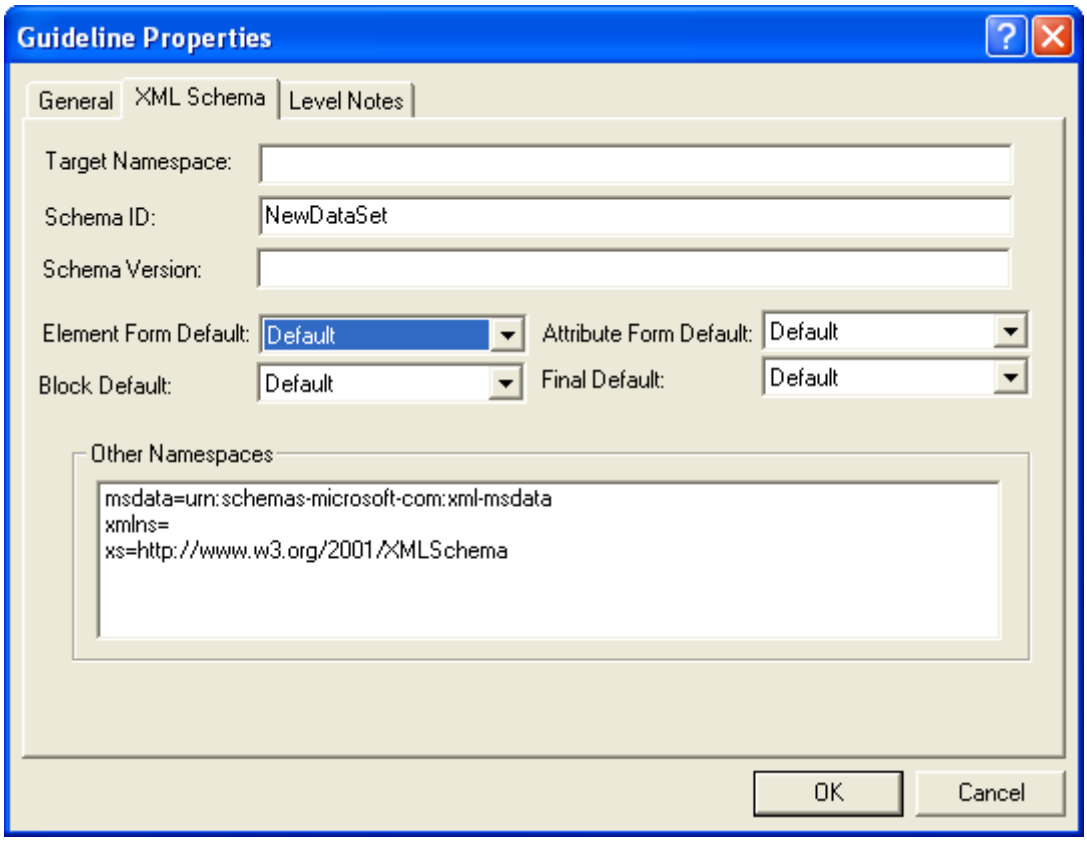
*Set Usage/Status
*Use Application Value List
*Use Code Set
ICD Business Rules (HIPAA Only)
Lookahead Business Rules

Properties

To set the properties for ax XML guideline, use one of these:

- Choose **File | Properties | XML Schema**
- Use the properties toolbar button: 
- Use the **Properties** button on the Save dialog box

The XML Schema tab looks like this:



Field	Contents
Target Namespace	Namespace for this schema. For your own use. TIBCO Foresight products do not use this field.
Schema ID	ID for this schema. For your own use. TIBCO Foresight products do not use this field.
Schema Version	Version for your own use. TIBCO Foresight products do not use this field.
Element Form Default	Determines the default Form setting for elements in your schema: must they be qualified by the target namespace? If set to Default in the Properties dialog box, then the default setting for XML will be used.
Attribute Form Default	Determines the default Form setting for attributes in your schema: must they be qualified by the target namespace? If set to Default here, then the default setting for XML will be used.
Block Default	Determines the default Block setting for your schema: elements with what derivations should be blocked from replacing this element? If set to Default here, then the default setting for XML will be used.
Final Default	Determines the default Final setting for your schema: how can this element be derived? If set to Default here, then the default setting for XML will be used.
Other Namespaces	Other namespaces used to qualify elements in this schema

Exporting a schema or DTD

To export the current schema to an .**xsd** file:

1. Save changes
2. Choose **File | Export | Export Current Guideline |To Schema** or **To DTD**
3. Choose the filename and location.

SEF

EDISIM stores guidelines in a public domain format called SEF in two ways:

- You can export an XML guideline in SEF format by using **Export | Export Current Guideline | To SEF**.
- When you import a schema or DTD into EDISIM, it is stored as a file with extension STD in EDISIM's **User Files\Public Guideline**s directory. This file is also in SEF format.

The SEF format that captures schema information has expanded from its earlier formats. If you use EDISIM's SEF files in other applications, please contact TIBCO Foresight technical support.

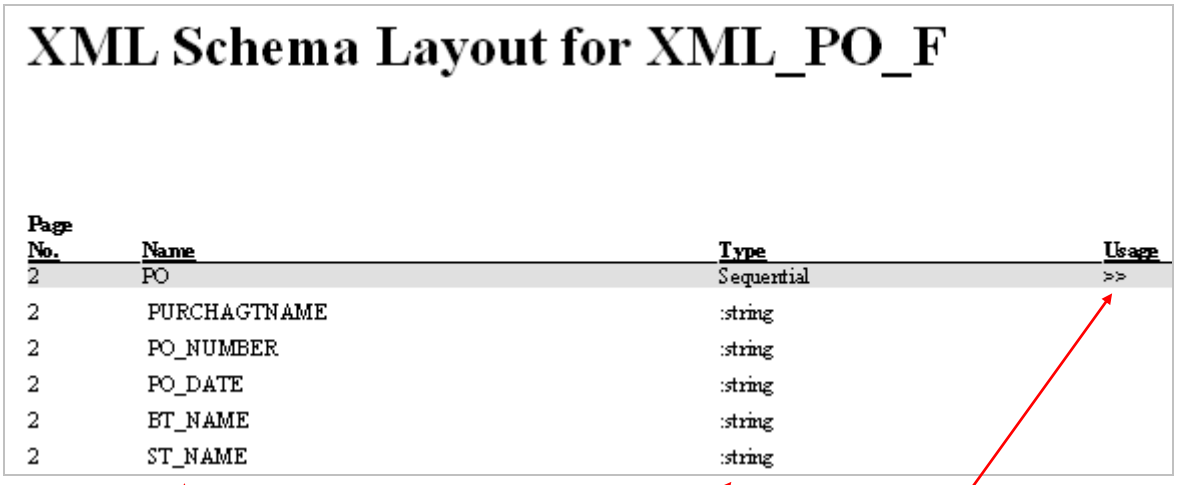
Printing DTDs and schemas

Printing formatted guideline documentation

EDISIM Document Builder will print formatted guideline documentation for XML schemas and DTDs that have been imported into Standards Editor.

Because of the nature of XML, the Doc Builder output has been adjusted and does not match Doc Builder EDI output exactly.

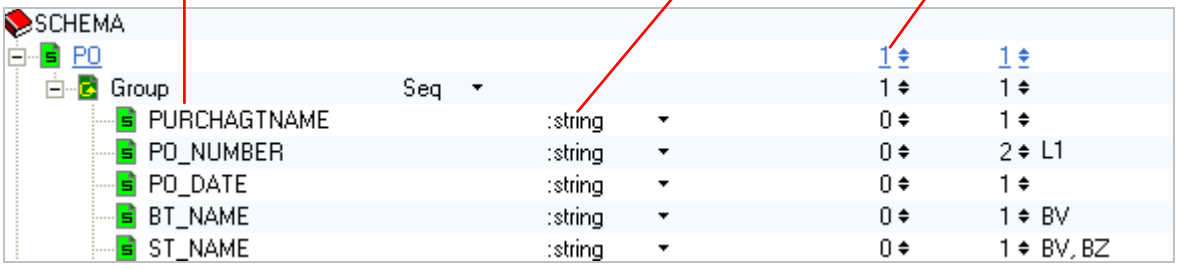
Doc Builder example page 1



XML Schema Layout for XML_PO_F

Page No.	Name	Type	Usage
2	PO	Sequential	>>
2	PURCHAGTNAME	:string	
2	PO_NUMBER	:string	
2	PO_DATE	:string	
2	BT_NAME	:string	
2	ST_NAME	:string	

Same guideline in Standards Editor



SCHEMA				
PO			1	1
Group	Seq		1	1
PURCHAGTNAME		:string	0	1
PO_NUMBER		:string	0	2 L1
PO_DATE		:string	0	1
BT_NAME		:string	0	1 BV
ST_NAME		:string	0	1 BV, BZ

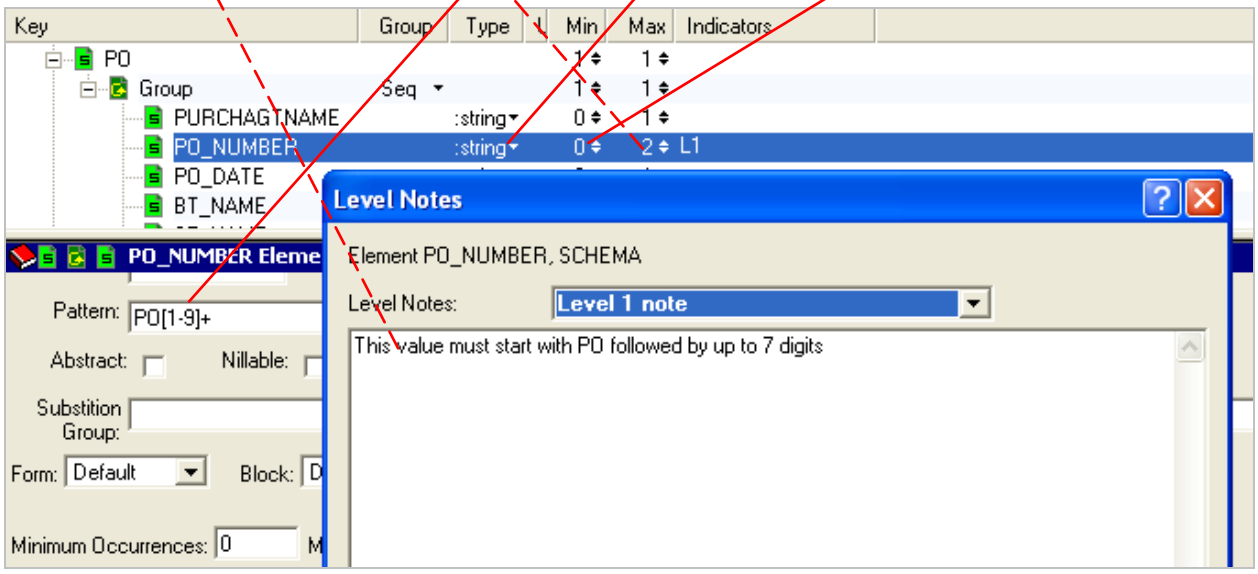
Doc Builder example page 2

Element: PO
Occurrences: 1

Attribute and Sub-Element Summary

Name	Description	Type	Usage
Child Elements and Groups			
PURCHAGTNAME	Purchase Order name Occurrences: None to 1, XML ID: PO_KC	:string	Optional
PO_NUMBER	Occurrences: None to 2, Length: / 9, Pattern: PO[1-9]+ This value must start with PO followed by up to 7 digits	:string	Optional

Same guideline in Standards Editor



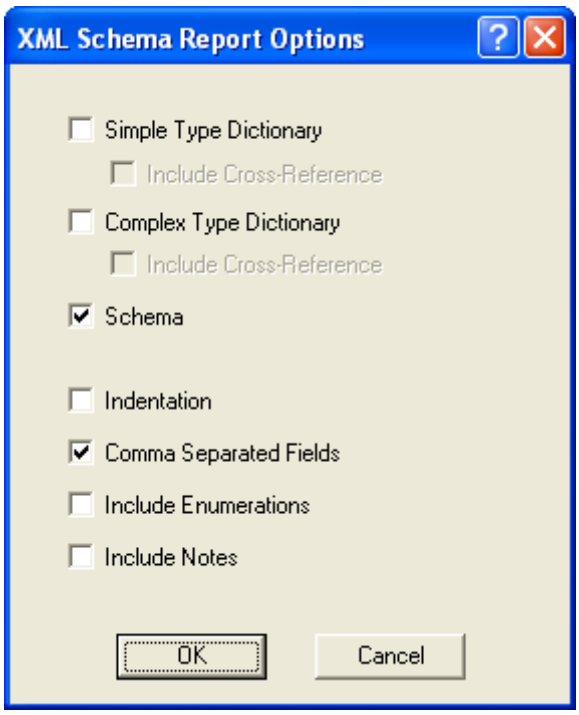
Many Doc Builder profile settings apply to XML guidelines.

Profile Tab	Setting
Formatting	All work with XML
Fonts	All work with XML
Usage Indicators	*Most do not apply to XML* Element indicator for Must Use affects the “segment table”
Shading	Shading works for Level Notes
Transaction Items	*Does not apply to XML*
Segment Items	*Does not apply to XML*
Element Items	*Does not apply to XML*
Segment Usage Notes	*Does not apply to XML*
Element Usage Notes	*Does not apply to XML*
Global Items	Level Notes

Printing a text report

You can print your current XML guideline in tabular or CSV format right from Standards Editor.

1. Open the schema.
2. Choose **File | Print**.
3. Choose the options that you want:
   - **Indentation** prints an indented tabular report.
   - **Comma Separated Fields** creates a CSV file suitable for importing into a spreadsheet or other application. If this is not selected, you will get a tabular text report.
   - The other selections determine the content of the output.



4. After clicking **OK**, you can use the **Save As…** and **Print** buttons at the bottom right.

Using EDISIM Validator with XML

Before you validate XML

For XML validation, you must first import the corresponding schema into EDISIM so that it will be put into the EDISIM database in the form of a STD file and become available during validation.

TIBCO Foresight programs that validate XML

These TIBCO Foresight products validate XML:

EDISIM Validator This program is the most convenient way to validate XML since it is installed as part of EDISIM 6.0 and later. It will validate your XML data without copying files anywhere.

HIPAA Validator Desktop
Copy the XML guideline's **STD** file from EDISIM's **User Files\Public Guidelines** directory to HIPAA Validator Desktop's **Database** directory. You can then HIPAA Validator Desktop and validate XML against that guideline.

Instream Copy the XML guideline's **STD** file from EDISIM's **User Files\Public Guidelines** directory to Instream's **Database** directory. You can then use Instream to validate XML against that guideline.

The rest of this section describes how to validate with EDISIM Validator.

Opening EDISIM Validator

To open EDISIM Validator, use one of these methods:

- From Standards Editor, click the toolbar icon for EDISIM Validator:
   or  or  (depending on EDISIM version).
- From the Start menu, select **Programs | FORESIGHT | EDISIM | Validator**.
- From Windows Explorer, go to EDISIM's **Bin** directory and double-click on **Validtr.exe**.

Validating XML data

You can validate XML with:

- EDISIM validator (data correction not available for XML)
- HIPAA Validator Desktop (data correction not available for XML)
- Instream (and optionally use trading partner automation - see **TIB_fsp-instream_*<n.n>*_tpa.pdf.**)

Validating with EDISIM Validator

1. Open EDISIM Validator.
2. Open an XML file:
   a. Choose **File | Open**.
   b. Navigate to the file. You can see some XML files in EDISIM's Samples directory.

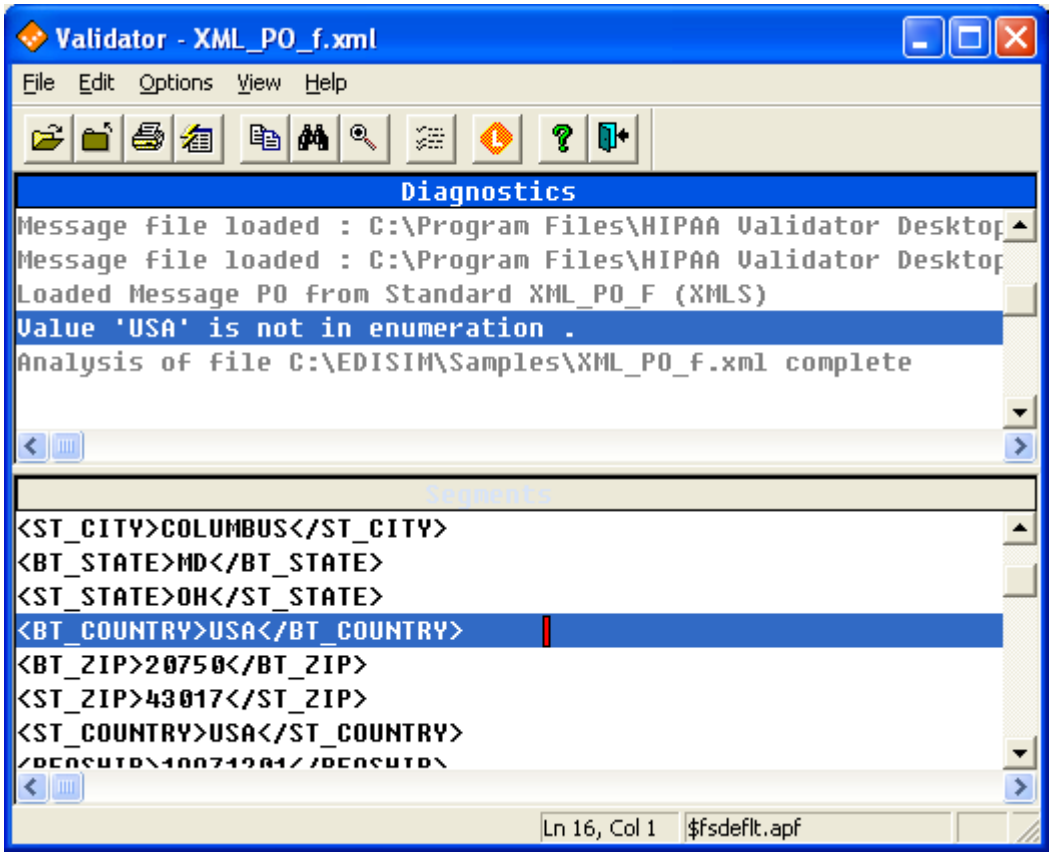
      As an XML example, you can choose **XML_PO_f.xml**.
3. Choose the XML guideline that defines how the data should look.

   If you are validating XML_PO_f, choose **XML_PO_F**.
4. Click **OK**.
5. Click **OK** when the analysis is complete.

You can also use trading partner automation to select guideline and profile for XML validations. See **TIB_fsp-instream_*<n.n>*_tpa.pdf**.

Viewing the results

The top pane contains diagnostic messages and the bottom pane contains the EDI segments or XML elements.



1. Click on the first line in the top pane.

   The corresponding line in the data is highlighted in the bottom pane.

2. Use your cursor keys to move down line by line in the top pane, noticing what happens in the bottom pane.

3. When you click on a blue line (an error), the bottom pane highlights the location of the error.

4. Continue scrolling down and looking at each message.

Creating an HTML report of XML validation results

You can create an HTML report from your XML validation. Please see **ValidationHighlighter.pdf**.

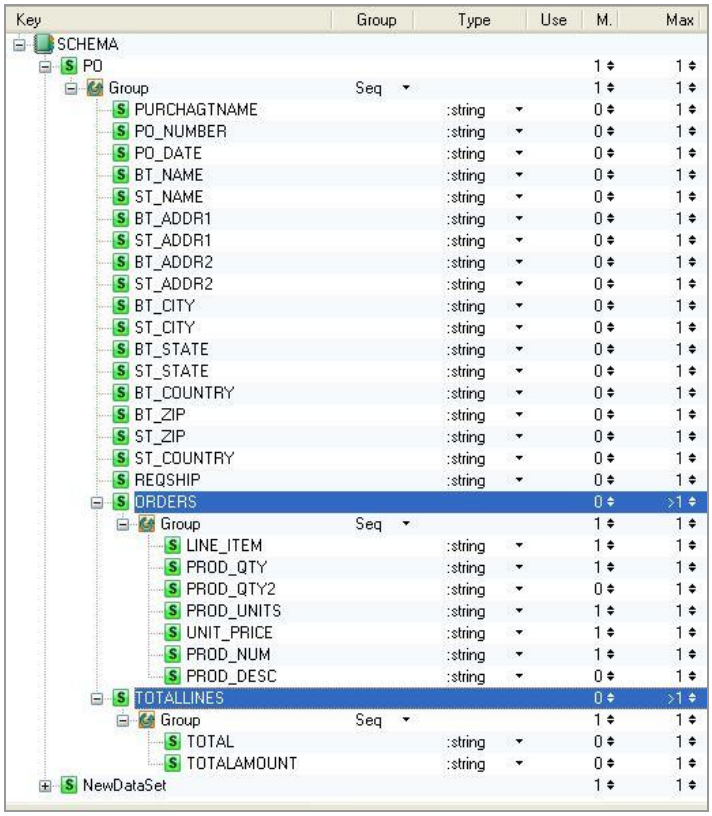
Splitting XML data

Overview

XML splitting is available on Windows platforms only.

Docsplitter on Windows can split XML at any complex element that repeats.

Example: you can split at ORDERS or TOTALLINES, which are complex elements and have Max of greater than 1.



XML splitting demo

V_DS_XML_split_PO in Instream's Scripts directory.

INI file for splitting XML data

XML splitting is controlled by the Docsplitter INI file.

[Split Point] section

Include a [Split Point] section that controls where the split will occur.

It looks like this:

```
[Split Point]
Xml=PO|PO/ORDERS
```

The format is:

[Split Point]

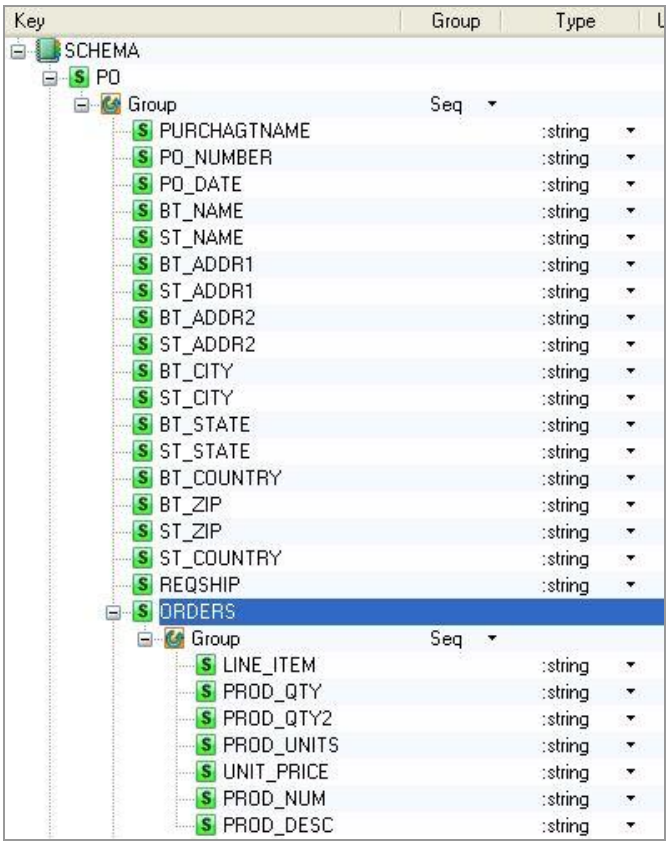
Xml=*rootelem* | *PathToSplit*

Where:

[Split Point]	Literal text
Xml=	Literal text
rootelem | *PathToSplit*	The root element followed by a vertical bar and then the path from the root element to the split point, with each element separated by a slash:

Example 1

The root element is PO and the split point is at the ORDERS complex element:



The INI file will contain this:

```
[Split Point]
Xml=PO|PO/ORDERS
```

This INI file will cause the XML data to split at the bold lines if an error is encountered within an ORDERS element:

```
<?xml version="1.0" standalone="yes"?>
<PO>
     <PURCHAGTNAME>ARTHUR JONES</PURCHAGTNAME>
     <PO_NUMBER>76766541</PO_NUMBER>
     <PO_DATE>19970905</PO_DATE>
     <REQSHIP>19971201</REQSHIP>
     <ORDERS>
          <LINE_ITEM>1</LINE_ITEM>
          <PROD_QTY>1000</PROD_QTY>
          <PROD_UNITS>EA</PROD_UNITS>
          <UNIT_PRICE>0.04</UNIT_PRICE>
          <PROD_NUM>2837645985</PROD_NUM>
          <PROD_DESC>BOOT EYELETS</PROD_DESC>
     </ORDERS>
```

```
<ORDERS>
     <LINE_ITEM>2</LINE_ITEM>
     <PROD_QTY>40</PROD_QTY>
     <PROD_UNITS>CA</PROD_UNITS>
     <UNIT_PRICE>54.00</UNIT_PRICE>
     <PROD_NUM>SAL3337465</PROD_NUM>
     <PROD_DESC>SOLE RESIN</PROD_DESC>
</ORDERS>
```

Example 2

The path is complex element LINEITEM within ORDERS:

The INI file will contain this:

```
[Split Point]
Xml=PO|PO/ORDERS/LINEITEM
```

[Summary Point] section

You can include an optional [Summary Point] section to display a count of the number of split points in the file.

The format is:

[Summary Point]

Xml="*element*"

Where:

[Summary Point]	Literal text.
Xml=	Literal text.
"<*element*>"	An element to hold the count.

Example 1

This INI file says the split point is the ORDERS element and the TOTAL element should contain a count of the ORDERS elements in the file. The TOTAL element here is reporting 10 ORDERS in the file:

INI file	Data
[Split Point] Xml=PO|PO/ORDERS [Summary Point] XML="<TOTAL>"	. . . <PROD_NUM>**4335625486**</PROD_NUM> <PROD_DESC>**PLASTIC STAKES**</PROD_DESC> </ORDERS> <ORDERS> <LINE_ITEM>**10**</LINE_ITEM> <PROD_QTY>**5000**</PROD_QTY> <PROD_UNITS>**EACH**</PROD_UNITS> <UNIT_PRICE>**0.29**</UNIT_PRICE> <PROD_NUM>**5548876542**</PROD_NUM> <PROD_DESC>**KAVER INSIGNIA PATCHES**</PROD </ORDERS> <TOTALLINES> <TOTAL>**10**</TOTAL> <TOTALAMOUNT>**186**</TOTALAMOUNT> </TOTALLINES> </PO>

Guideline and APF changes for splitting XML data

No changes are needed to the XML guideline or the validation profile (APF).

Docsplitter command line for splitting XML data

The command line is similar to the Docsplitter command line for splitting EDI data, with these considerations:

-i The **i** parameter is required and must point to a validation detail results file that was created by validating with an XML guideline.

-s The **s** parameter is required and must point to a Docsplitter INI file that contains, as a minimum, a [Split Point] section with an Xml line. See INI file for splitting XML on page 46.

For an example, please see V_DS_XML_split_PO in Instream's Scripts directory.

For details about the Docsplitter command line, see **TIB_fsp-instream_*<n.n>*_docsplitter.pdf**.