TIBCO ActiveMatrix BusinessWorks™ Plug-in for HL7 User's Guide

Software Release 7.3

March 2016

Two-Second Advantage®

TIBCO®

Important Information

Contents

TIBCO Documentation and Support Services

All TIBCO documentation is available on the TIBCO Documentation site, which can be found here:

https://docs.tibco.com

Product-Specific Documentation

The following documents for this product can be found on the TIBCO Documentation site:

- *TIBCO ActiveMatrix BusinessWorks™ Plug-in for HL7 Installation* - Read this manual for instructions on site preparation and installation.
- *TIBCO ActiveMatrix BusinessWorks™ Plug-in for HL7 Release Notes* - Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for this release.
- *TIBCO ActiveMatrix BusinessWorks™ Plug-in for HL7 User's Guide* - Read this manual for instructions on using the product and the shipped samples.

Other TIBCO Product Documentation

When working with the TIBCO ActiveMatrix BusinessWorks™ Plug-in for HL7, you may find it useful to read the documentation for the following TIBCO products:

- TIBCO ActiveMatrix BusinessWorks™
- TIBCO Foresight® EDISIM®
- TIBCO Foresight® Instream®
- TIBCO Foresight® Translator

How to Contact TIBCO Support

For comments or problems with this manual or the software it addresses, contact TIBCO Support as follows:

- For an overview of TIBCO Support, and information about getting started with TIBCO Support, visit this site:

  http://www.tibco.com/services/support

- If you already have a valid maintenance or support contract, visit this site:

  https://support.tibco.com

  Entry to this site requires a user name and password. If you do not have a user name, you can request one.

How to Join TIBCOmmunity

TIBCOmmunity is an online destination for TIBCO customers, partners, and resident experts. It is a place to share and access the collective experience of the TIBCO community. TIBCOmmunity offers forums, blogs, and access to a variety of resources. To register, go to:

http://www.tibcommunity.com

HL7 Plug-in Overview

TIBCO ActiveMatrix™ BusinessWorks is an easy to use integration product suite for enterprise, web, and mobile applications. It uses the Eclipse graphical user interface (GUI) for defining business processes and the process engine to execute them. TIBCO ActiveMatrix BusinessWorks™ Plug-in for HL7 plugs into TIBCO ActiveMatrix™ BusinessWorks, and provides HL7 message customization together with business process design and management.

TIBCO ActiveMatrix™ BusinessWorks supports plug-ins that extend the palette functionality. After installing the plug-in, an HL7 Palette becomes available in TIBCO Business Studio™. You can add the plug-in activities to the business processes you are designing, and integrate them into the process flow. At runtime, the plug-in activities are executed as part of the TIBCO ActiveMatrix™ BusinessWorks process execution.

The plug-in allows users to:

- translate HL7 data
- validate HL7 data
- parse HL7 header information
- generate acknowledgements
- send and receive HL7 messages using Lower Level Protocol (LLP).

Getting Started

A typical workflow using the TIBCO ActiveMatrix™ BusinessWorks Plug-in for HL7 includes creating a project, designing a process, and deploying the application.

TIBCO ActiveMatrix™ BusinessWorks enables users to create services and integrate applications, and deploy them at runtime. It uses the Eclipse graphical user interface (GUI) for defining business processes and the process engine to execute them.

To design a process and deploy it at runtime, refer to the following:

1. Creating a Project
2. Designing a Process
3. Testing an Application
4. Deploying an Application

See *TIBCO ActiveMatrix BusinessWorks™ Application Deployment* for more detailed information.

Creating a Project

The first task of using the plug-in is creating a project. After creating a project, you can add resources and processes.

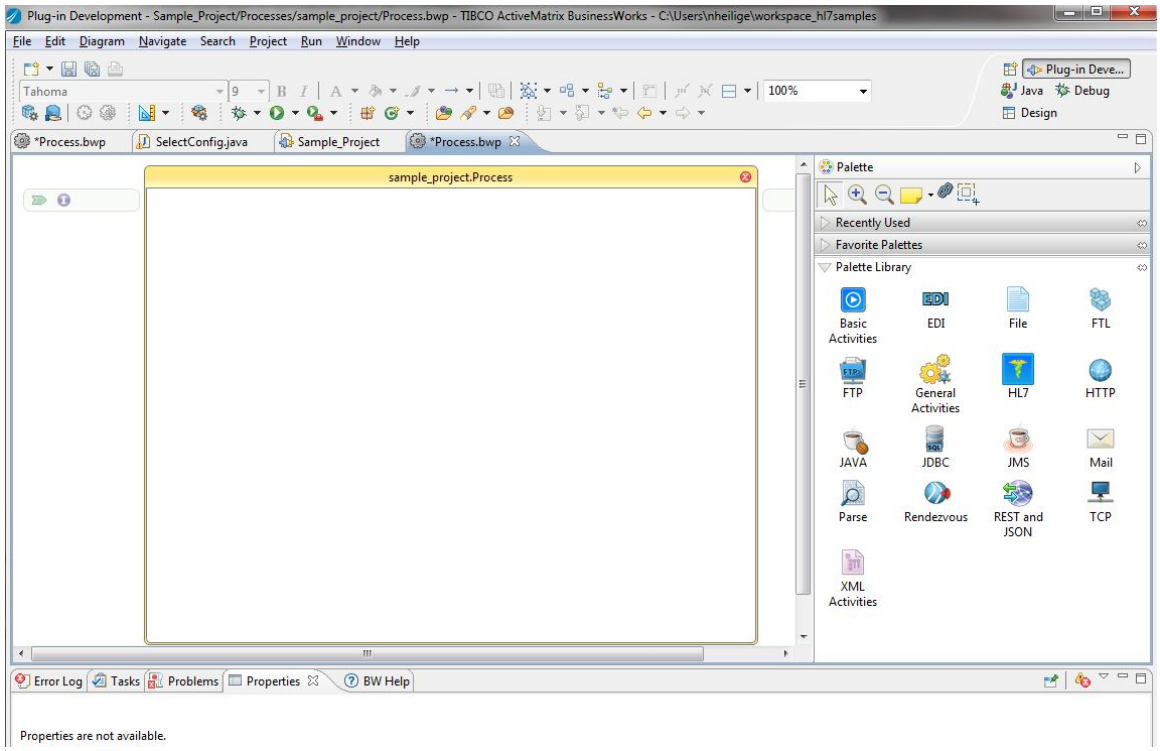
An Eclipse project is an application module configured for TIBCO ActiveMatrix BusinessWorks. An application module is the smallest unit of resources that is named, versioned, and packaged as part of an application.

Procedure

1. Start TIBCO Business Studio:
   - Click **Start** > **All Programs** > **TIBCO** > *TIBCO_HOME* > **TIBCO Business Studio** *version_number* > **Studio for Designers**.
2. From the menu, click **File** > **New** > **BusinessWorks Resources** to open the BusinessWorks Resource Wizard.
3. In the "Select a wizard" dialog, click **BusinessWorks Application Module** and click **Next** to open the New BusinessWorks Application Module wizard.
4. In the Project dialog, configure the project that you want to create:
   a) In the **Project name** field, enter a project name.
   b) By default, the created project is located in the workspace current in use. If you do not want to use the default location for the project, clear the **Use default location** check box and click **Browse** to select a new location.
   c) Use the default version of the application module, or enter a new version in the **Version** field.
   d) Keep the **Create empty process** and **Create Application** check boxes selected to automatically create an empty process and an application when creating the project.
   e) Select the **Use Java configuration** check box if you want to create a Java module.
      A Java module provides the Java tool capabilities.
   f) Click **Finish** to create the project.

Result

The project with the specified settings is displayed in the Project Explorer view.



Designing a Process

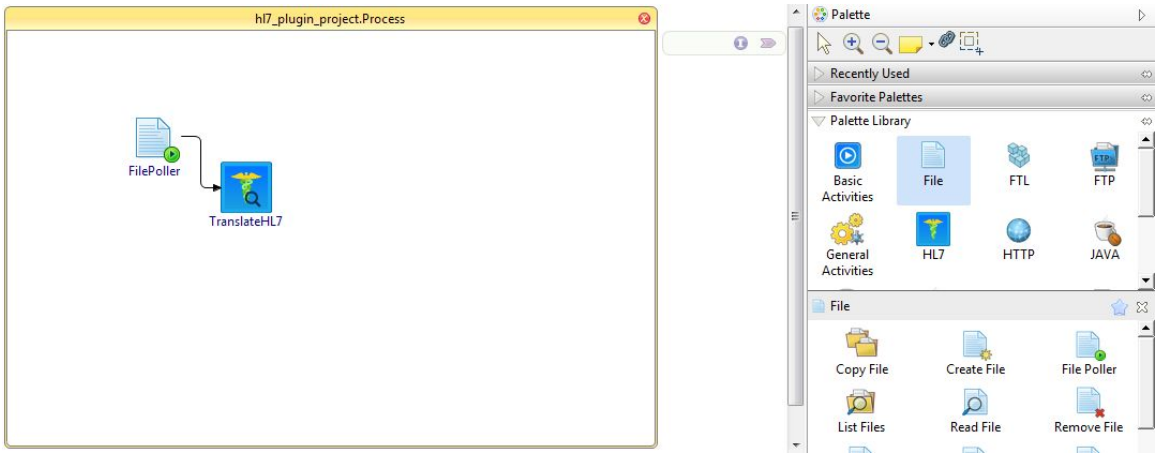
Processes capture and manage the flow of business information in an enterprise between different data sources and destinations. You can design a process by using activities and adding conditions.

By default, an empty process is created when creating a project with the **Create empty process** check box selected.

See *TIBCO ActiveMatrix™ BusinessWorks Application Deployment* for more details about creating processes.

Procedure

1. In the Process editor, select and drop an activity from the Palette view.
   For example, select and drop the FilePoller activity from the **File** palette and the TranslateHL7 activity from the **HL7** palette.



You can also select activities from the **Context** menu.

2. Click the  icon to create links between the activities.
3. Configure the added activities.
4. Click **File** > **Save** to save the process.

Testing an Application

An application contains an application module that is defined in TIBCO Business Studio™. After you design your business process, you can run and debug the configurations.

TIBCO Business Studio™ has a built-in debugger that allows users to debug the design-time configurations.

Prerequisites

Ensure that you have created an application and designed a process before testing.

By default, a process and an application are created when creating a project with the **Create empty process** and **Create Application** check boxes selected.

See *TIBCO ActiveMatrix BusinessWorks™ Application Deployment* for more details about creating processes and applications.

Procedure

1. Click **Run** > **Run** to run the application.
2. Optional: Click **Run** > **Debug** to debug the application.
   The perspective changes to Debug from Modeling. You can view the job details in the Console view from the Debug perspective.

Deploying an Application

You can manage TIBCO ActiveMatrix™ BusinessWorks applications by using TIBCO® Enterprise Administrator after deploying the applications.

Prerequisites

An enterprise archive (EAR) file must be generated before deploying an application. See Generating an EAR File for details about generating an EAR file.

A complete workflow of deployment includes:

1. Building an EAR file.
2. Uploading the EAR file.
3. Deploying the EAR file.
4. Starting the application.

You can deploy an application EAR file from TIBCO Business Studio™, or by using the command-line mode with the **bwdesign** utility. See *TIBCO ActiveMatrix BusinessWorks™ Administration* for more details about how to deploy an application.

Generating an EAR File

Application archives are the enterprise archive (EAR) files that are created in TIBCO Business Studio™. An EAR file is required when deploying an application.

Prerequisites

An application project has already been created, as explained in Creating a Project.

Procedure

1. Go to File Explorer and click the Open Directory to Browse  icon.
2. Select the folder where you want to generate the EAR file and click **OK**.
   The new folder is displayed in the File Explorer view.
3. Drag the application from the Project Explorer to the new folder in the File Explorer.

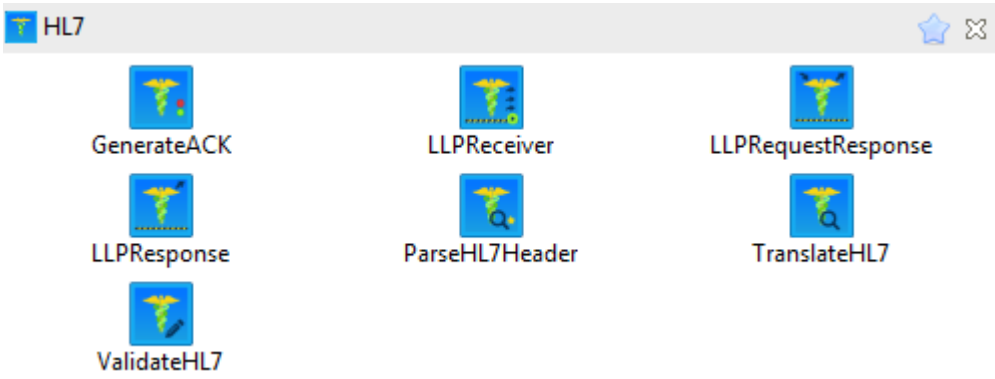
   The EAR file is generated with the name `<application>_<version>.ear`.

HL7 Palette Overview

The HL7 Palette contains HL7 activities for TIBCO ActiveMatrix BusinessWorks™ Plug-in for HL7.

One of the key activities in integrating health care systems is to implement the Interface Engine functionality using the TIBCO ActiveMatrix BusinessWorks™ Process Engine in the course of defining process models. This process modeling is done in TIBCO Business Studio™ using the HL7 features, as well as the many features that are part of TIBCO ActiveMatrix BusinessWorks™. These include FTP, the file poller, and logging support.

The HL7 palette, found on the Palettes panel, includes seven activities.



To achieve different functionalities, HL7 activities can be divided into the following groups.

Basic activities

The following activities are most likely to be used in your HL7 integration project:

- TranslateHL7
- ValidateHL7
- ParseHL7Header
- Generate Acknowledgement

LLP activities

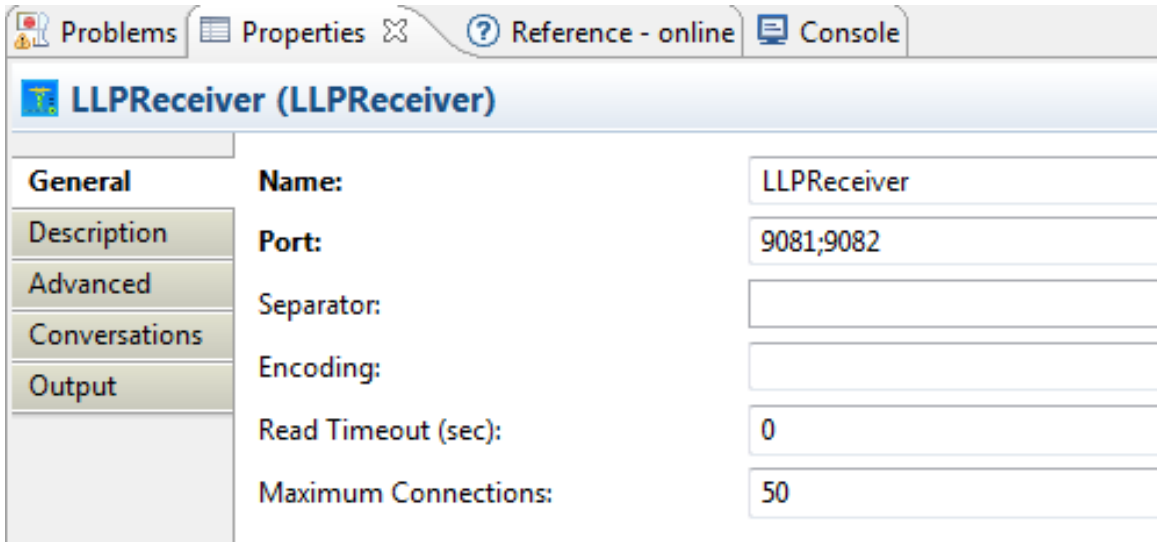
Most HL7 messages are sent using TCP over Lower Level Protocol (LLP). The following activities are related to TCP over LLP and are used to send or receive HL7 messages to or from other systems in the provider space.

- LLP Receiver
- LLPRequestResponse
- LLPResponse

Configuring Activities

When an activity has been added to the workspace, the activity can be configured by clicking on the activity icon.

The properties view area below the workspace is populated with fields associated with the activity. Each activity has configuration tabs on the left, and a corresponding configuration input area on the right.



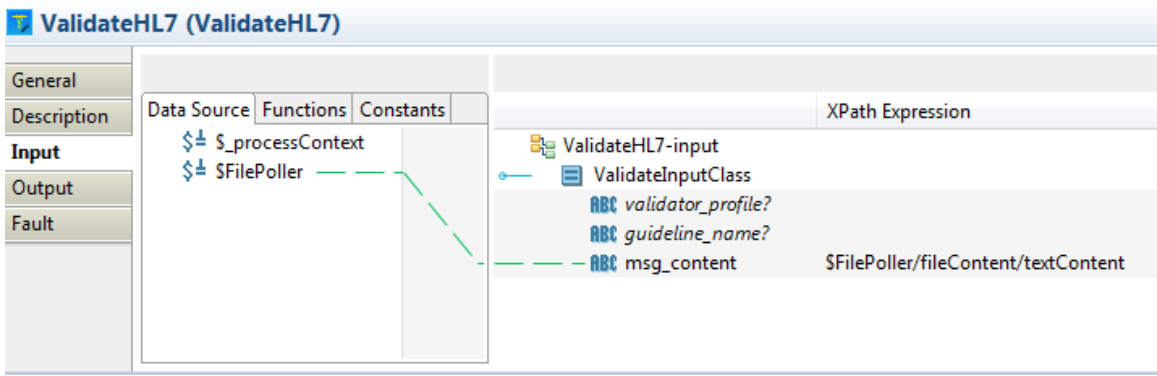
The configuration properties are grouped under tabs such as General, Description, Input, Output, and so on. For example, upon adding an LLPReceiver activity, you can configure it by specifying the values for the properties under the tabs General, Description, Advanced, Conversations, and Output.

See Working with Standard Activity Features in the *TIBCO ActiveMatrix BusinessWorks™ Application Development* document for details.

Configuration Using XPath

Input, Output, Advanced, and Fault information is configured and displayed using the XPath Builder. XPath uses path expressions to navigate through XML documents. XPath also has basic manipulation functions for strings, numbers, and booleans.

TIBCO ActiveMatrix BusinessWorks™ uses XPath as the language for defining conditions and transformations.



See XPath in the *TIBCO ActiveMatrix BusinessWorks™ Application Development* document for details.

GenerateACK

Use the GenerateACK activity to generate acknowledgements for input HL7 messages.

General

The **General** tab contains the following fields.

Field	Literal Value/ Module Property/ Process Property?	Description
Name	No	The name to be displayed as the label for the activity in the process.

Description

The **Description** tab is used to provide a short description for the activity.

Input

The **Input** tab contains the following fields.

Field	Datatype	Description
hl7Message	string	The incoming HL7 message for generating an ACK or NAK message.
defaultMSHwhenHeaderInvalid	string	If the MSH segment of the input HL7 message is invalid, use this field to specify a valid MSH segment for the input HL7 message. If you do not provide a valid MSH segment, the following is used as a default value: `MSH|^~\\&||||||ACK||P|2.3||  ||` This field takes effect only if the MSH segment of the input HL7 message is invalid.
msaTextMsg	string	The text in the MSA segment of the input HL7 message. The default value for this field is null (""). Override the default by specifying text here.

Field	Datatype	Description
msaAckCode	string	The acknowledgment (ACK) code in the MSA segment of the input HL7 message. The default value for this field is AE. Override the default by specifying an ACK code here. The following acknowledgment codes can be used: AA, AE, AR, A, E, R, accept, error, reject, CA, CE, and C. When msaAckCode is set to AA, A, or "accept," output item ackIsNak will output as false.

Output

Output Item	Datatype	Description
ackContent	string	The generated acknowledgment message.
ackIsNak	boolean	Indicates whether the generated acknowledgment is a NAK or an ACK message. If the value is true the generated acknowledgment is a NAK message. If the value is false the generated acknowledgment is an ACK message. The following ACK codes in the output MSAsegment can be used: AA, A, CA, and accept.

Fault

The **Fault** tab lists exceptions that are thrown by this activity.

Error Schema Element	Datatype	Description
msg	string	Error message description.
msgCode	string	The error code. It represents TIBCO ActiveMatrix BusinessWorks™ Plug-in for HL7 and TIBCO Foresight® Translator and TIBCO Foresight® Instream® errors. See Error Codes for details.

LLPReceiver

Use the LLPReceiver activity to receive HL7 messages on one port.

Examples

Refer to the associated examples:

- Receive Process

General

The **General** tab contains the following fields.

Field	Literal Value/ Module Property/ Process Property?	Description
Name	No	The name to be displayed as the label for the activity in the process.
Port	Yes	The port through which your client connects to a server system. Only one port can be given.
Separator	Yes	List of allowed separators: Minimal LLP - Default Linefeed Linefeed-Linefeed Carriage return-linefeed Empty line using linefeed Empty line using Carriage return-linefeed No separator **Note** The separator only separates messages. It does not separate a message into lines.
Encoding	Yes	The encoding used to convert between binary and text formats.
Read Timeout (sec)	Yes	Number of seconds before server terminates the connection.
Maximum Connections	Yes	Maximum number of simultaneous sessions that can connect with the server. The default is 20.

Description

The **Description** tab is used to provide a short description for the activity.

Advanced

The **Advanced** tab contains the following fields.

Field	Description
Sequence Key	This field can contain an XPath expression that specifies which processes should run in a specified order. Process instances with sequencing keys evaluating to the same value are executed sequentially in the order of the process instance creation. See *TIBCO ActiveMatrix BusinessWorks™ Application Development* for more information about controlling the execution order of process instances and about XPath expressions.
Custom Job Id	This field can contain an XPath expression that specifies a custom ID for the process instance.

Conversations

In the **Conversations** tab, you can initiate or join a conversation.

Conversations receive a message after creating a process instance, which is initiated or joined by an activity. This activity can be used to initiate a conversation. Click the **Add New Conversation** icon to initiate a conversation. For details about conversations, see *TIBCO ActiveMatrix BusinessWorks™ Application Development*.

Input

There are no dynamic inputs for this activity.

Output

The **Output** for the activity can be the incoming HL7 message string.

LLPRequestResponse

Use the LLPRequestResponse activity to send a request and receive a synchronous response. The request and response can be in the form of an HL7 string or XML data.

Synchronous response means that the response is sent on the same TCP connection. If a connection is closed, and a response is sent on a different TCP connection, then it is called an asynchronous response.

This activity sends its request to the LLP Receiver activity, which passes the request to the LLP Response activity. If the Is One Way checkbox is not checked, the LLP Request Response activity may get an empty string if the LLP Receiver activity is suspended before the LLP Response activity returns the response to the LLP Receiver.

Examples

Refer to the associated examples:

- Illustrate AL Process
- Illustrate NE Process

General

The **General** tab contains the following fields.

Field	Literal Value/ Module Property/ Process Property?	Description
Name	No	The name to be displayed as the label for the activity in the process.
Host	Yes	The machine name or IP of the server machine.
Port	Yes	The port through which your client connects to a server system. Only one port can be given.
Separator	Yes	List of allowed separators: Minimal LLP - Default Linefeed Linefeed-linefeed Carriage return-linefeed Empty line using linefeed Empty line using Carriage return-linefeed No separator  The separator only separates messages. It does not separate a message into lines.
Encoding	Yes	The encoding used to convert between binary and text formats.
Request Timeout (msec)	Yes	Number of milliseconds to wait before the request or read times out. Default is 0.
Connection Retries	Yes	Number of retries if connection fails. Default is 1.
Retry wait (sec)	Yes	Number of seconds to wait before next retry, if any. Default is 5.
Is One Way	Yes	If checked, after sending a request, a response is not expected. If unchecked, after sending a request, a response is expected.

Description

The **Description** tab is used to provide a short description for the activity.

Input

The **Input** tab contains the following fields.

Input Item	Datatype	Description
data	string	The request HL7 message string to be sent out.

Input Item	Datatype	Description
encoding	string	Optional. Encoding information about the message to be sent out.
separator	string	Optional. You can specify a special message separator here.

Output

The output for the activity can be the incoming HL7 message string.

Fault

The **Fault** tab lists exceptions that are thrown by this activity.

Error Schema Element	Datatype	Description
msg	string	Error message description.
msgCode	string	The error code. It represents TIBCO ActiveMatrix BusinessWorks™ Plug-in for HL7 and TIBCO Foresight® Translator and TIBCO Foresight® Instream® errors. See Error Codes for details.

LLPResponse

Use the LLPResponse activity to send a synchronous response back to the initiating system on TCP over LLP.

This activity can work in combination with the LLPReceiver activity.

Examples

Refer to the associated examples:

- Receive Process

General

The **General** tab contains the following fields.

Field	Literal Value/ Module Property/ Process Property?	Description
Name	No	The name to be displayed as the label for the activity in the process.

Field	Literal Value/ Module Property/ Process Property?	Description
Reply for	No	Provides a choice of LLP Receivers in the process to be used.
Separator	Yes	List of allowed separators: Minimal LLP - Default Linefeed Linefeed-Linefeed Carriage return-linefeed Empty line using linefeed Empty line using Carriage return-linefeed No separator **Note** The separator only separates messages. It does not separate a message into lines.

Description

The **Description** tab is used to provide a short description for the activity.

Input

The **Input** tab contains the following fields.

Field	Datatype	Description
data	string	The request HL7 message string to be sent out.
encoding	string	Optional. Encoding information about the message to be sent out.
separator	string	Optional. You can specify a special message separator here.

Fault

The **Fault** tab lists exceptions that are thrown by this activity.

Error Schema Element	Datatype	Description
msg	string	Error message description.
msgCode	string	The error code. It represents TIBCO ActiveMatrix BusinessWorks™ Plug-in for HL7 and TIBCO Foresight® Translator and TIBCO Foresight® Instream® errors. See Error Codes for details.

ParseHL7 Header

Use the ParseHL7 Header activity to parse the header of a set of HL7 string messages and generate acknowledgements.

The acknowledgement is generated as a part of the output message; it returns the following information:

- If the header of the input HL7 string message is valid, the first field of the MSA segment in the acknowledgement message specifies the acknowledge status: AA.
- If the header of the input HL7 string message is invalid, this activity will not generate an acknowledgement message and will throw an exception. To generate an acknowledgement message, you need to design a proper workflow.

Examples

Refer to the associated examples:

- Receive Process

General

The **General** tab contains the following fields.

Field	Literal Value/ Module Property/ Process Property?	Description
Name	No	The name to be displayed as the label for the activity in the process.

Description

The **Description** tab is used to provide a short description for the activity.

Input

The input for the activity are HL7 messages. This can contain one or more HL7 messages.

Output

The **Output** tab contains the following fields.

Output Item	Datatype	Description
Messages	string	*message* The HL7 message. *ackMsg* The auto-generated acknowledgement for the incoming message. *responseID* The value of the second field in the MSA segment of the incoming message. This MSA segment is present only if the incoming message is a response message. This field is optional; the request message does not have an MSA segment.
Header	string	All the header field values.

Fault

The **Fault** tab lists exceptions that are thrown by this activity.

Error Schema Element	Datatype	Description
msg	string	Error message description.
msgCode	string	The error code. It represents TIBCO ActiveMatrix BusinessWorks™ Plug-in for HL7 and TIBCO Foresight® Translator and TIBCO Foresight® Instream® errors. See Error Codes for details.

ValidateHL7

The ValidateHL7 activity utilizes TIBCO Foresight® Instream® to validate an EDI file against a specified guideline and provides summary and a detail output containing the results.

Use the ValidateHL7 activity to validate HL7 input data and generate a result and a summary report.

Examples

Refer to the associated examples:

- Validate and Translate Example

General

The **General** tab contains the following fields.

Field	Literal Value/ Module Property/ Process Property?	Description
Name	No	The name to be displayed as the label for the activity in the process.
CallBack	No	Select this checkbox to use a Java Class for callback (optional). See ValidateHL7 Callback for more complete information on Enabling Callbacks.
Class Name	No	Specify a Java Class. This field is enabled when CallBack is checked.  Create a new Java Class.  Browse to an existing Java Class using the Type selection dialog. See ValidateHL7 Callback for more complete information on Enabling Callbacks.

Description

The **Description** tab is used to provide a short description for the activity.

Input

The **Input** tab contains the following fields.

Field	Datatype	Description
validator_profile	string	(Optional) Contains configuration information for this activity. Each time you validate, this file is read to determine what is to be checked and what is to be displayed on the screen or written to output files. The value is the full path of the validator_profile file. The `$fsdeflt.apf` file, which is installed in the TIBCO Foresight® Instream® bin directory, is the default profile file for the validate operation. If the `$fsdeflt.apf` file is modified, the changes do not take effect until you stop and restart TIBCO Business Studio™.
guideline_name	string	The name of the guideline used for the validation. It must be in the `TIBCO_FORESIGHT_HOME /Instream/Database` directory. If you want to share guidelines with the `TIBCO_FORESIGHT_HOME / Translator/Database` directory, refer to the Share Guidelines and Maps section below.
msg_content	string	The input data for validation.

Share Guidelines and Maps

If you install TIBCO Foresight® Translator and TIBCO Foresight® Instream® in different directories, you can share their Database directories by modifying the `TIBCO_FORESIGHT_HOME /Instream/bin/$dir.ini` or `TIBCO_FORESIGHT_HOME /Translator/bin/$dir.ini` file.

For example, if you want to use the `TIBCO_FORESIGHT_HOME /Instream/Database` directory as shared, perform the following steps:

1. Open the `TIBCO_FORESIGHT_HOME /Translator/bin/$dir.ini` file.
2. Modify the database path and point it to the `TIBCO_FORESIGHT_HOME /Instream/Database` directory.

   The following is an example:

```
:* FORESIGHT Supplied .STD Standard Files
FSFACTORY ="C:\Foresight\Instream\Static"

:*User-defined .STD Standard Files
ALLUSERSSHARED ="C:\Foresight\Instream\Database"
```

   The value must be enclosed with double quotation marks.

3. After making the modification, you can put the involved XSD, STD, and MAP files either in the `TIBCO_FORESIGHT_HOME /Translator/Database` directory or in the `TIBCO_FORESIGHT_HOME /Instream/Database` directory.

Output

The result and the summary report are listed in the Output tab. For details about the output items, see the *Instream Validation Technical Manual*.

The **Output** tab contains the following fields.

Output Item	Datatype	Description
Validation	string	*Validation_result_in_flatfile* Specifies the message of the validation detail results in Flat File format. *Validation_summary_in_flatfile* Specifies the message of the validation summary results in Flat File format.

Output Item	Datatype	Description
Validation_summary	numeric	*Severity* Error severity summary record. *Type* Error type summary record, containing the following counts: ignoreCount infoCount warningCount errorCount fatalCount user1Count user2Count
total_of_errors	numeric	Total number of errors.
return_code	numeric	Specifies the TIBCO Foresight® Instream® return code. For example, 100 means the validation ran successfully, 133 means the database directory cannot be opened, and so on. Return codes are listed in the TIBCO Foresight® Instream® documentation.

Fault

The **Fault** tab lists exceptions that are thrown by this activity.

Error Schema Element	Datatype	Description
msg	string	Error message description.
msgCode	string	The error code. It represents TIBCO ActiveMatrix BusinessWorks Plug-in for HL7 and TIBCO Foresight errors. See Error Codes for details.

ValidateHL7 CallBack

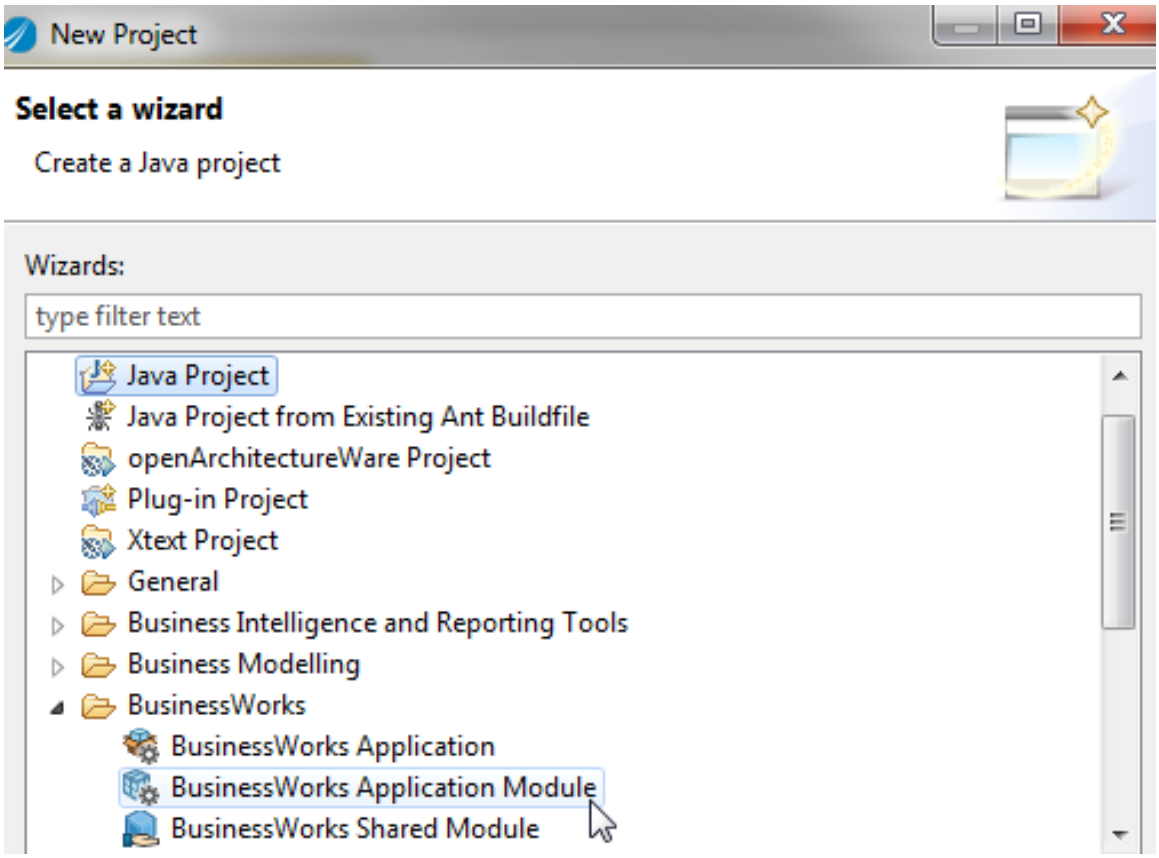
Overview

The ValidateHL7 CallBack allows you to select validation guidelines and profiles based on the contents of the input data by modifying the Java code.

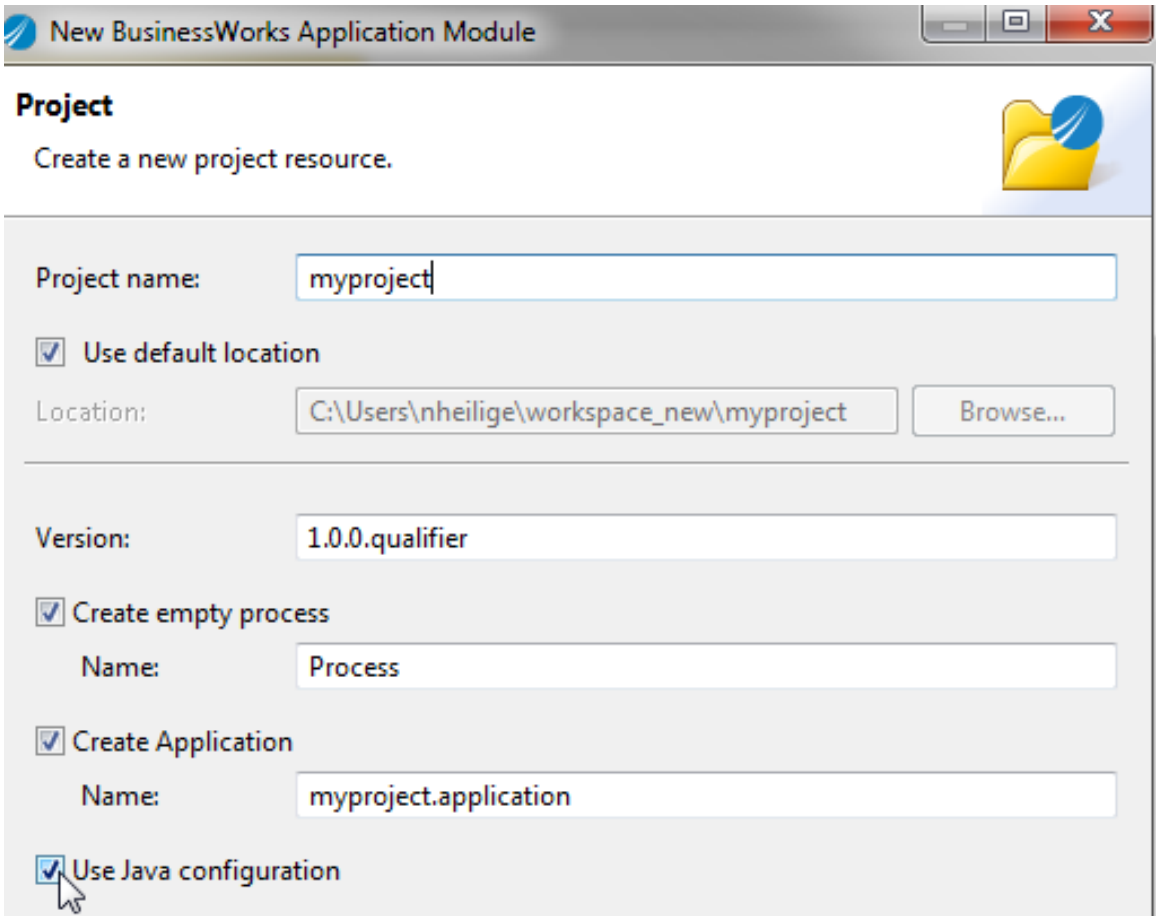
Enable Callback

To enable the ValidateHL7 CallBack:

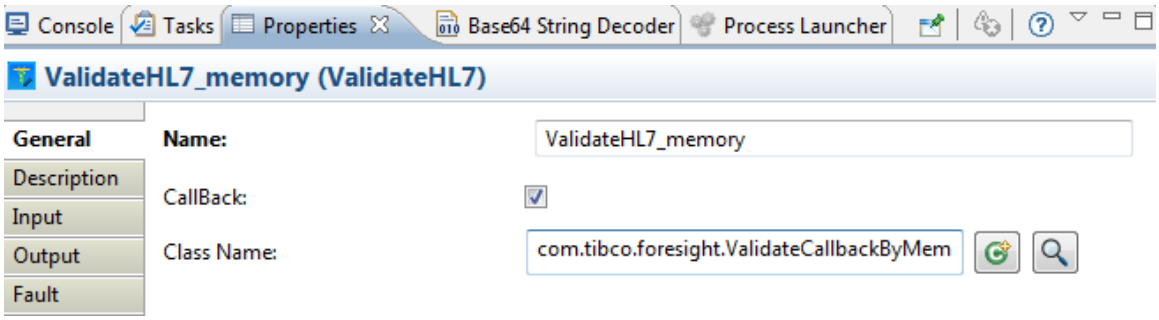
1. Select the TIBCO ActiveMatrix BusinessWorks™ Application Module when creating a new project.



2. Enter a Project name, select the **Use Java configuration** checkbox, and click **Next**.

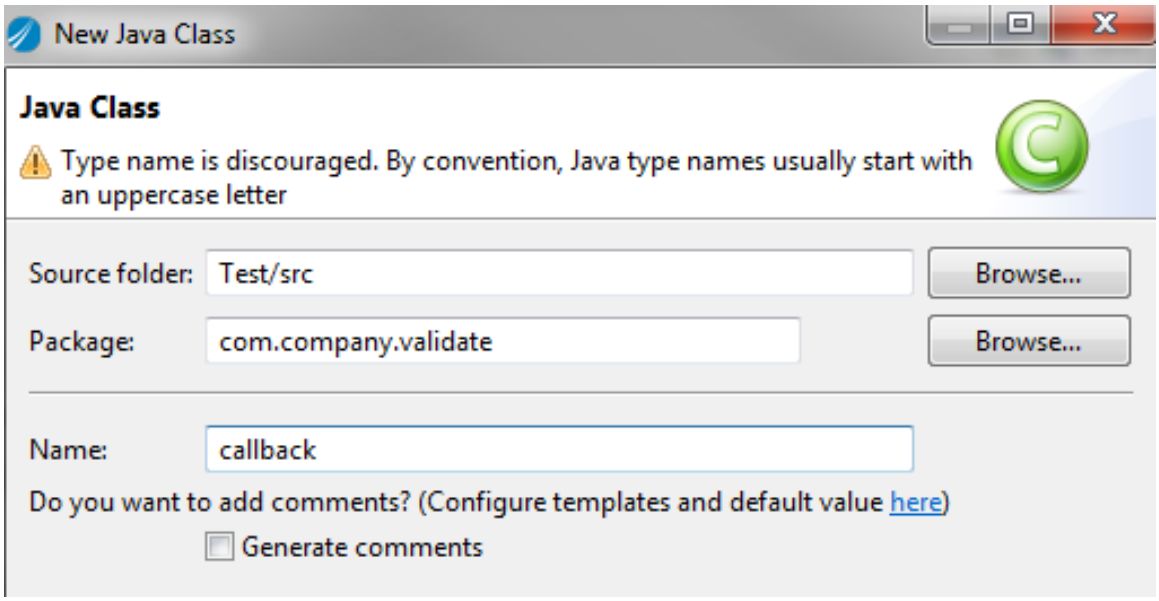


3. Select the **CallBack** checkbox in the ValidateHL7 activity General tab and click the **Create a new class** icon  .



4. Fill in Class Name and click **Finish**.

   A Java class is created with default functions and variables.



Result

The following MSH information will set to `HashMap<String, String> infoMap`.

```
key = Info.ReceivingApplicationUniversalID
key = Info.SendingFacilityNamespaceID
key = Info.MessageTypeMessageStructure
key = Info.MessageHeader, value = MSH
key = Info.SendingFacilityUniversalID
key = Info.ReceivingFacilityUniversalID
key = Info.id, value = HL7
key = Info.SendingFacilityUniversalIDType
key = Info.ReceivingFacilityUniversalIDType
key = Info.SendingApplicationNamespaceID
key = Info.DateTimeofMessage
key = Info.ReceivingApplicationNamespaceID
key = Info.ReceivingApplicationUniversalIDType
key = Info.SendingApplicationUniversalID
key = Info.Version
key = Info.ReceivingFacilityNamespaceID
key = Info.MessageTypeTriggerEvent
key = Info.SendingApplicationUniversalIDType
key = Info.MessageTypeMessageCode
```

Use the “get” function to access the values in the map.

```
infoMap.get("Info.MessageTypeMessageCode")
```

Setting Guidelines and Profiles

You can set guidelines and profiles by modifying `public void selectGuideline()` in the Java code.

Refer to the associated examples:

- Validate and Translate Example

If Guideline Resides on Disk (Instream Database Directory)

Sign the map image to variable `mapFileBuffer`.

```
guidelineName = “VXR_V03.std”;
```

If Guideline Resides in Memory (Database)

1. Sign the guideline image to variable `guidelineByteBuffer`.

   ```
   guidelineByteBuffer=(guideline in byte[]);
   ```

2. Sign the guideline name to variable `guidelineName`.

```
guidelineName = “VXR_V03.std”;
```

Setting a Profile

Directly assign the profile name to the variable `apfFileName`.

```
apfFilename=”user_profile.apf”;
```

TranslateHL7

The TranslateHL7 activity utilizes TIBCO Foresight® Translator to translate a file from one format to another format.

Use the Translate HL7 activity to specify a translation type and translation map to be used to translate input data from one format to another with no interim staging required.

Examples

Refer to the associated examples:

- Validate and Translate Example

Translated Data Formats

Translation for the following data formats is supported by this activity.

From (Source)	To (Target)
HL7	XML
HL7	HL7
HL7	Flat File
XML	HL7
Flat File	HL7

General

The **General** tab contains the following fields.

Field	Literal Value/ Module Property/ Process Property?	Description
Name	No	The name to be displayed as the label for the activity in the process.

Field	Literal Value/ Module Property/ Process Property?	Description
Operation Type	No	This drop-down list allows you to specify the operation type. Refer to the Translated Data Formats section above.
CallBack	No	Select this checkbox to use a Java Class for callback (optional).  This function is not available for XML input. See to TranslateHL7 Callback for more complete information on Enabling Callbacks.
Class Name	No	Specify a Java Class. This field is enabled when CallBack is checked.  Create a new Java Class.  Browse to an existing Java Class using the Type selection dialog. See to TranslateHL7 Callback for more complete information on Enabling Callbacks.

Description

The **Description** tab is used to provide a short description for the activity.

Input

The **Input** tab contains the following fields.

Field	Datatype	Description
map_filename	string	This field specifies the name and extension of a map file. TIBCO ActiveMatrix BusinessWorks™ Plug-in for HL7 supports map files with the .map and .xml file name extensions. A map file specifies the source and target guidelines for each translation map. The source guideline describes the data before translation. The target guideline describes the data after translation.  All involved XSD, STD, and MAP files must be in the `TIBCO_FORESIGHT_HOME /Translator/Database` directory. If you want to share guidelines with the `TIBCO_FORESIGHT_HOME /Instream/Database` directory, refer to the Share Guidelines and Maps section below.
msg_content	string	The message content to be translated.

Field	Datatype	Description
input encoding	string	(Optional) The input encoding used for translation. This is used when translating to ISO-8859-1 only. Valid values are blank (no encoding) or ISO-8859-1.
output encoding	string	(Optional) The output encoding used for translation. Valid values are blank (no encoding), UTF-8, UTF-16, or ISO-8859-1.

Share Guidelines and Maps

If you install TIBCO Foresight® Translator and TIBCO Foresight® Instream® in different directories, you can share their Database directories by modifying the `TIBCO_FORESIGHT_HOME /Instream/bin/$dir.ini` or `TIBCO_FORESIGHT_HOME /Translator/bin/$dir.ini` file.

For example, if you want to use the `TIBCO_FORESIGHT_HOME /Instream/Database` directory as shared, perform the following steps:

1. Open the `TIBCO_FORESIGHT_HOME /Translator/bin/$dir.ini` file.
2. Modify the database path and point it to the `TIBCO_FORESIGHT_HOME /Instream/Database` directory.

   The following is an example:

```
:* FORESIGHT Supplied .STD Standard Files
FSFACTORY ="C:\Foresight\Instream\Static"

:*User-defined .STD Standard Files
ALLUSERSSHARED ="C:\Foresight\Instream\Database"
```

   The value must be enclosed with double quotation marks.

3. After making the modification, you can put the involved XSD, STD, and MAP files either in the `TIBCO_FORESIGHT_HOME /Translator/Database` directory or in the `TIBCO_FORESIGHT_HOME /Instream/Database` directory.

Output

The **Output** tab contains the following fields.

Output Item	Datatype	Description
msg_content	string	The output of this activity is the translated message in the specified format, which can be the HL7, XML, or Flat File format. The generated message format is specified in the Operation Type field on the Configuration tab.
return_code	numeric	Specifies the TIBCO Foresight® Translator return code. For example, 100 means the transaction ran successfully, 158 means the map file cannot be opened, and so on. Translator return codes are listed in the TIBCO Foresight® Translator documentation.
encoding	string	The encoding used for translation.

Fault

The **Fault** tab lists exceptions that are thrown by this activity.

Error Schema Element	Datatype	Description
msg	string	Error message description.
msgCode	string	The error code. It represents TIBCO ActiveMatrix BusinessWorks™ Plug-in for HL7 and TIBCO Foresight® Translator errors. See Error Codes for details.

TranslateHL7 CallBack

Overview

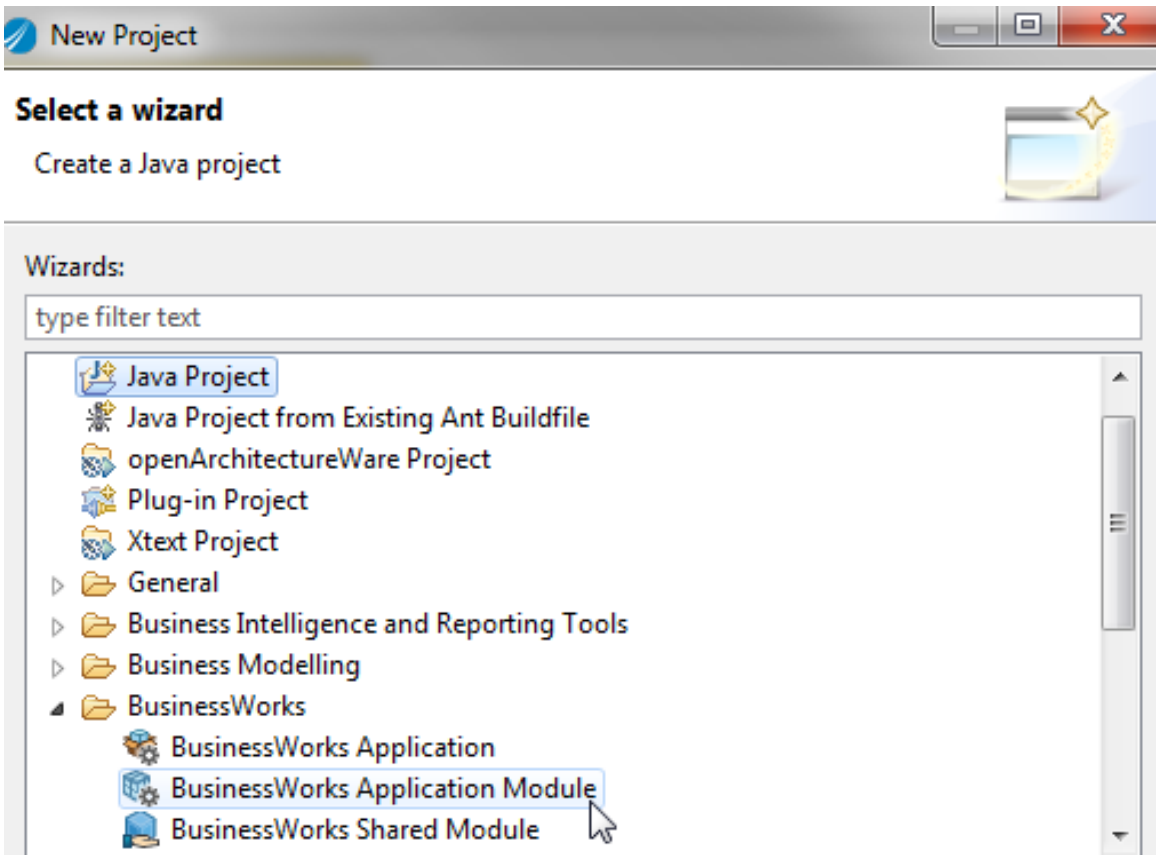
TranslateHL7 CallBack allows you to select a translation map based on the contents of the input data by modifying the Java code.

This function is not available for XML input.

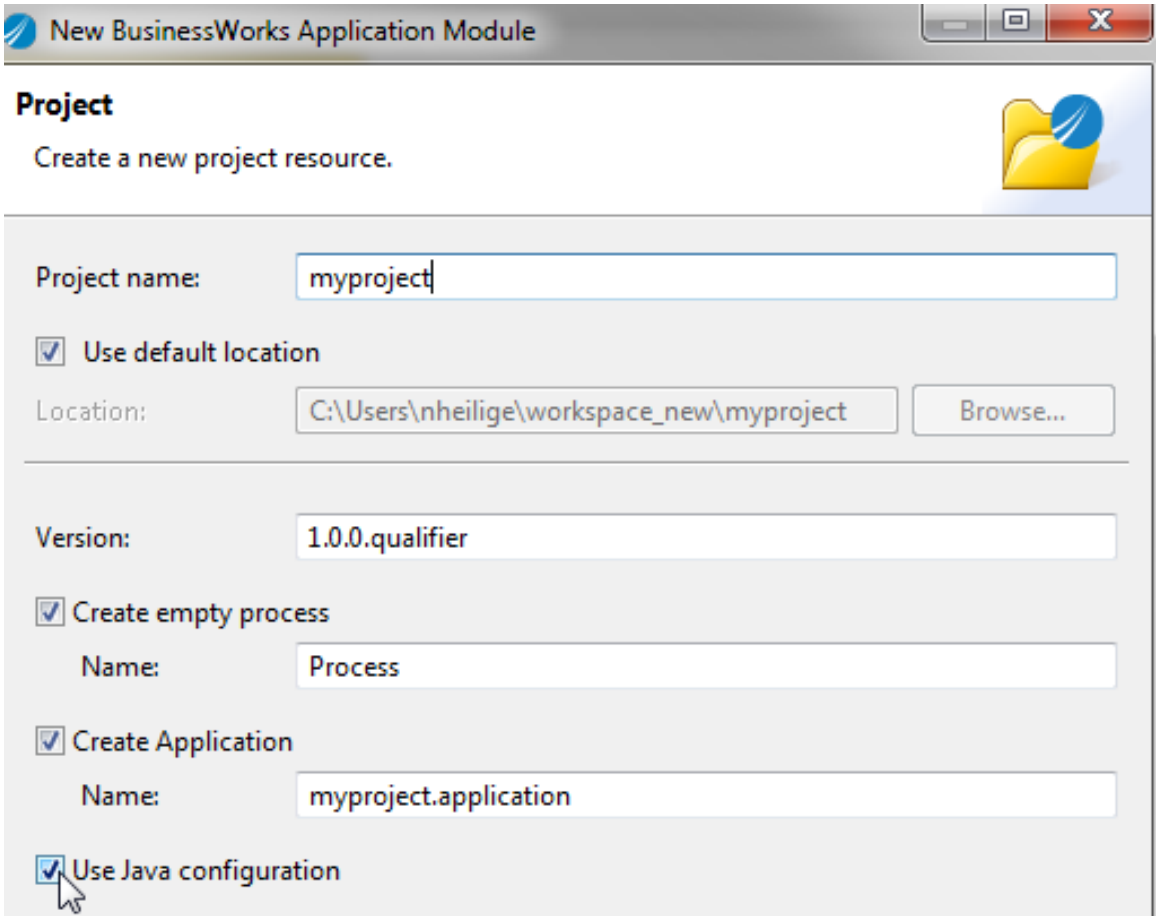
Enable Callback

To enable TranslateHL7 CallBack:

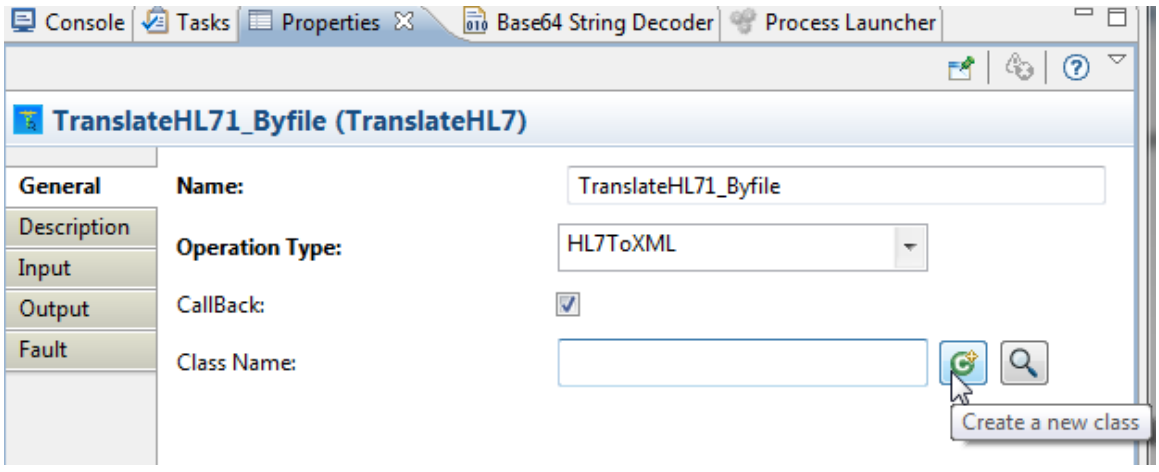
1. Select the TIBCO ActiveMatrix BusinessWorks™ Application Module when creating a new project.



2. Enter a project name, select the **Use Java** configuration checkbox, and click **Next**.

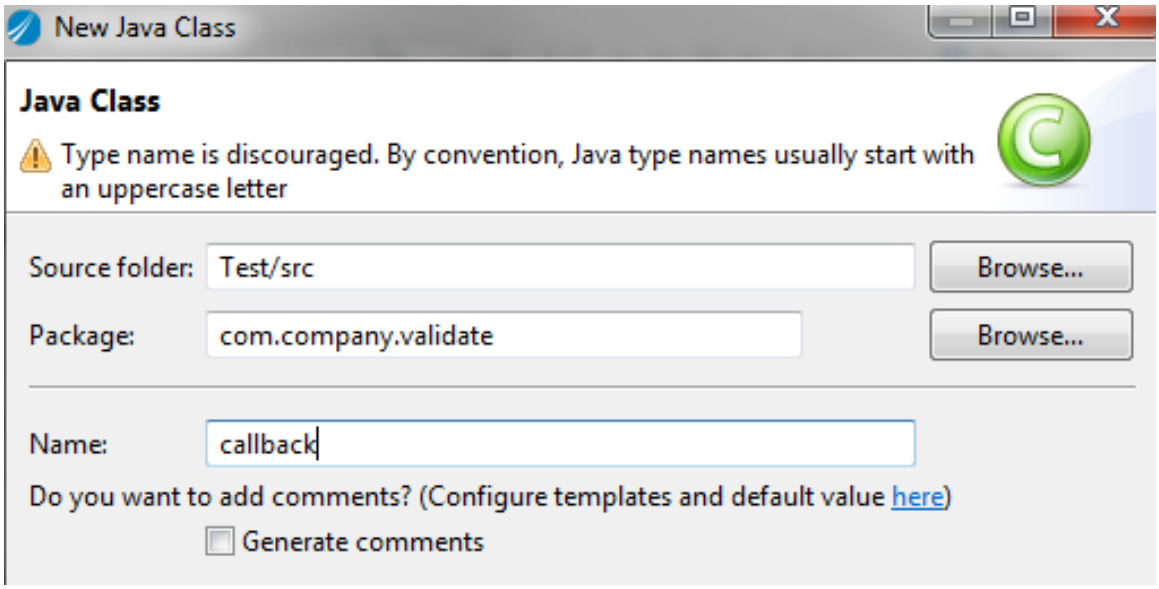


3. Select the **CallBack** checkbox in the TranslateHL7 activity General tab and click the **Create a new class** icon  .



4. Enter a class name and click **Finish**.

   A Java class is created with default functions and variables.



Result

The following (HL7 input) MSH information will be set to `HashMap<String, String> infoMap`

```
key = Info.ReceivingApplicationUniversalID
key = Info.SendingFacilityNamespaceID
key = Info.MessageTypeMessageStructure
key = Info.MessageHeader, value = MSH
key = Info.SendingFacilityUniversalID
key = Info.ReceivingFacilityUniversalID
key = Info.id, value = HL7
key = Info.SendingFacilityUniversalIDType
key = Info.ReceivingFacilityUniversalIDType
key = Info.SendingApplicationNamespaceID
key = Info.DateTimeofMessage
key = Info.ReceivingApplicationNamespaceID
key = Info.ReceivingApplicationUniversalIDType
key = Info.SendingApplicationUniversalID
key = Info.Version
key = Info.ReceivingFacilityNamespaceID
key = Info.MessageTypeTriggerEvent
key = Info.SendingApplicationUniversalIDType
key = Info.MessageTypeMessageCode
```

Use the "get" function to access the values in the map.

```
infoMap.get("Info.MessageTypeMessageCode")
```

Setting the Map, Source Guideline and Target Guideline

Refer to the associated examples:

- Validate and Translate Example

If Map Resides on Disk (Translator Database Directory)

Set up the Map file name.

1. Directly assign the map name to variable `mapFileName`.

   ```
   mapFileName = "VXR_V03_VXR_V03_EX.map";
   ```

2. Update the configName variable with the format `configName=mapname||`.

   ```
   configName="VXR_V03_VXR_V03_EX.map||"
   ```

If Map Resides in Memory (Database)

1. Update the configName variable with the format `configName = MapName|SourceGuidelineName|TargetGuidelineName;`
2. Assign the map image to variable `mapFileBuffer =(map image in byte[])`.
3. Specify the source guideline variable with the format `sourceGuidelineBuffer = (source guideline image in byte[]);`
4. Specify the target guideline variable with the format `targetGuidelineBuffer = (target guideline image in byte[]);`

HL7 Custom Functions

An important part of any HL7 integration project is the data mapping from one message format to another. TIBCO ActiveMatrix BusinessWorks™ offers advanced functionality in data mapping, including drag-and-drop and a comprehensive suite of built-in functions.

However, no matter how complete the built-in functions are, there is always a need to customize, especially in the healthcare industry. TIBCO ActiveMatrix BusinessWorks™ Plug-in for HL7 includes a useful set of custom functions.

See the TIBCO ActiveMatrix BusinessWorks™ documentation for detailed information on how to write, load, and use your own custom functions.

Using HL7 Custom Functions

TIBCO ActiveMatrix BusinessWorks™ Plug-in for HL7 includes a set of HL7-related custom functions, which are specified in the `bwpluginhl7_HOME/resources/HL7CustomFunctions.class` file.

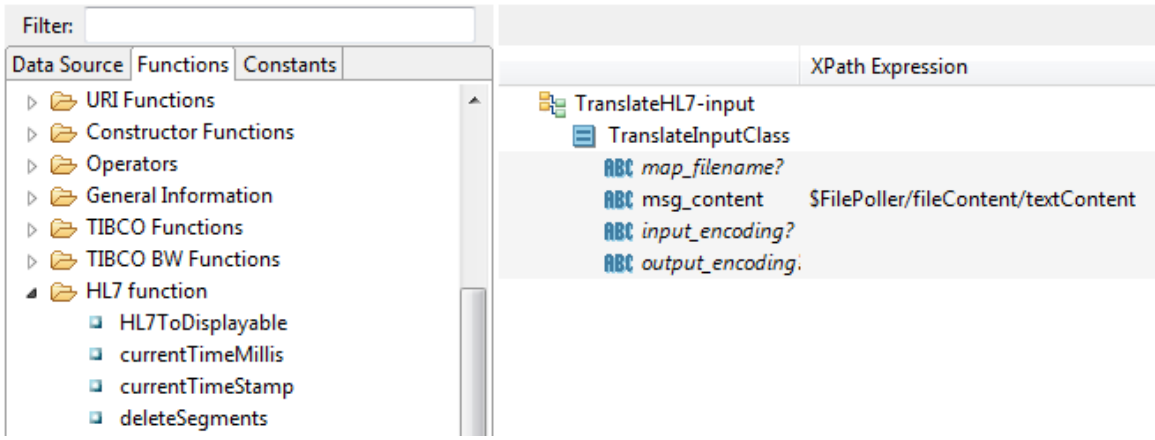
To use these functions:

1. Create or open a project in TIBCO Business Studio™.
2. Load the pre-defined class file into the project using the Java Custom Function resource.

Custom Functions Usage

To see the HL7 specific custom functions:

1. Click on an Activity on the process editor.
2. In the properties pane, select the appropriate tab (usually the Input tab) and click the **Functions** tab in the XPath Formula Builder dialog.
3. Expand the HL7 Functions folder. The custom HL7 functions appear.



Available HL7 Custom Functions

Function	Description and Sample Usage
HL7ToDisplayable	Replaces \r with system line separator. **Sample usage:** `HL7ToDisplayable($hl7Msg)`
currentTimeMillis	Returns current time in milliseconds.
currentTimeStamp	Returns current timestamp in HL7 format: yyyyMMddhhmmss.
deleteSegments	Returns HL7 message after the specified segments are removed. **Sample usage:** `deleteSegment($hl7Msg, "ZSH:EVN")` **Sample output:** Message will have EVN and ZSH segments removed
displayableToHL7	HL7 requires that all segments end with \r . This function converts DOS/ UNIX format file into HL7 ER7 format. **Sample usage:** `displayableToHL7($hl7MsgInDisplayFormat)`
extractField	Extracts the indexed field from HL7 Message from the first segment with a given segment name. **Sample usage:** `extractField($Hl7MsgString, $segmentName,$index )`
generateNewGUID	Generates Unique ID.
separateMsgsFromBatch	Returns individual messages in the batch by removing FHS/BHS/BTS/FTS segments.
trimEmptyFields	Trims empty fields from an HL7 message.

Managing Logs

Logs are used to trace and troubleshoot exceptions. The plug-in allows users to set up log levels and export logs. Additionally, the plug-in also allows users to enable TIBCO ActiveMatrix BusinessWorks™ Plug-in for HL7 logging.

- See Managing Plug-in Logs for more details about setting up log levels and exporting logs.
- See Enabling TIBCO HL7 Plug-in Logging for more details about enabling TIBCO ActiveMatrix BusinessWorks™ Plug-in for HL7 logging.

Managing Plug-in Logs

The plug-in allows users to set up log levels and export logs by modifying the `logback.xml` file.

TIBCO Business Studio™ provides a built-in debugger, which allows users to debug a project. When starting debugging or running a project, the logs are displayed in the Console view. You can set up the log level to define the log amount. To set up the console log level, see Setting Up Log Level.

The plug-in also allows users to export logs of the defined log levels to a file. To export logs to a file, see Exporting Logs.

Exporting Logs

You can set up the log level and export logs to a file by modifying the `logback.xml` file.

Procedure

1. Navigate to the *TIBCO_HOME*`\bw\6.3\config\design\logback` directory and open the `logback.xml` file.

   When deploying the application in TIBCO® Enterprise Administrator, you need to navigated to the *TIBCO_HOME*`\bw\domains\mydomain\appnodes\myspace\mynode` directory to find the `logback.xml` file.

2. Add the following node to specify the file location.

```
<appender name="FILE" class="ch.qos.logback.core.FileAppender">
   <file>c:/bw6-asbw.log</file>
      <encoder>
        <pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{36}-%msg%n</pattern>
      </encoder>
</appender>
```

   The `file` tag defines the location to which the log is exported and the value is the absolute path of the file that is detailed to the file name.

3. Add the following node to specify the log level.

```
<logger name="com.tibco.bw.palette.hl7">
   <appender-ref ref="FILE"/>
   <level value="Error"/>
</logger>
```

   The `level` tag defines the log level and the value can be `Error` or `Debug`.

4. Save the file.

Setting Up Log Level

When you run a process in TIBCO Business Studio™, the runtime logs are displayed in the Console view. You can set up the log level before running a process.

Procedure

1. Navigate to the *TIBCO_HOME*\bw\6.3\config\design\logback directory and open the logback.xml file.

   When deploying the application in TIBCO® Enterprise Administrator, you need to navigate to the *TIBCO_HOME*\bw\domains\mydomain\appnodes\myspace\mynode directory to find the logback.xml file.

2. Add the following node to specify the log level.

```
<logger name="com.tibco.bw.palette.hl7">
   <level value="Error"/>
</logger>
```

   The level tag defines the log level and the value can be Error or Debug.

3. Save the file.

Enabling TIBCO HL7 Plug-in Logging

The TIBCO ActiveMatrix BusinessWorks™ Plug-in for HL7 allows users to enable TIBCO HL7 Plug-in logging to troubleshoot errors that are thrown by TIBCO HL7 Plug-in.

Before deploying the application in TIBCO® Enterprise Administrator, you need to add the Log Properties listed in the Properties Added in the INI File column to the config.ini file to enable TIBCO HL7 Plug-in logging.

The config.ini file is located in the *TIBCO_HOME*\bw\domains\mydomain\appnodes\myappspace\myappnode directory.

Procedure

1. Start TIBCO Business Studio™.
2. Click **Run** > **Run Configurations**.
3. Click **BusinessWorks Application** > **BWApplication** in the left panel.
4. Click the **(x)=Arguments** tab in the right panel.
5. Type the following properties in the **VM arguments** panel. Click **Apply**.

```
-Dcom.tibco.plugin.as.filelog.enable=True
-Dcom.tibco.plugin.as.filelog.level=FINEST
-Dcom.tibco.plugin.as.filelog.directory=d:/logs/Loging_0107
-Dcom.tibco.plugin.as.filelog.filename=FINEST.log
```

   See Log Properties for more details about the value of each property.

Log Properties

The log properties allow users to set up log levels, specify the log file name, and the location of the log file.

The properties in the following columns are used when:

- Properties Added in VM Arguments

  you want to enable TIBCO ActiveMatrix BusinessWorks™ Plug-in for HL7 logging in TIBCO Business Studio™ before running a project.

- Properties Added in the INI File

  you want to enable TIBCO ActiveMatrix BusinessWorks™ Plug-in for HL7 logging in TIBCO Enterprise Administrator® before deploying an application.

See Enabling TIBCO HL7 Plug-in Logging for more details.

Properties Added in VM Arguments	Properties Added in the INI File	Datatype	Description
`-Dcom.tibco.plugin.as.filelog.enable`	`com.tibco.plugin.as.filelog.enable`	boolean	Set the value to `true` to enable TIBCO HL7 Plug-in logging.
`-Dcom.tibco.plugin.as.filelog.level`	`com.tibco.plugin.as.filelog.level`	string	The amount of logging produced by TIBCO HL7 Plug-in core library can be adjusted and retrieved using this property. The following values can be used: • `None`: Do not return any information. • `FATAL`: Return only fatal errors. • `ERROR`: Return errors. • `WARN`: Return warnings. • `INFO`: Return debug information. • `FINE`: Return fine debug information. • `FINER`: Return more detailed debug information. • `FINEST`: Return the most detailed debug information. The default value is `ERROR`.
`-Dcom.tibco.plugin.as.filelog.directory`	`com.tibco.plugin.as.filelog.directory`	string	The output destination of the log file.
`-Dcom.tibco.plugin.as.filelog.filename`	`com.tibco.plugin.as.filelog.filename`	string	The file name.

Error Codes

The exceptions that are thrown by the TIBCO ActiveMatrix BusinessWorks™ Plug-in for HL7 are listed with the corresponding descriptions and resolutions.

Error Code and Message	Role	Category	Description	Resolution
TIBCO-BW-PALETTE-HL7-501000 `The map is required.`	errorRole	BW-Plug-in ERROR_TRANSLATOR	A translation map is required.	Specify a map to be used for translation.
TIBCO-BW-PALETTE-HL7-501001 `The input data is required.`	errorRole	BW-Plug-in ERROR_TRANSLATOR	Input data is required.	Specify input data to be used for translation.
TIBCO-BW-PALETTE-HL7-501002 `The operation type is required.`	errorRole	BW-Plug-in ERROR_TRANSLATOR	An operation type is required.	Specify an operation type for translation. For example, hl7toXML, HL7toFLAT, etc.
TIBCO-BW-PALETTE-HL7-501004 `TIBCO-BW-PALETTE-HL7-501004: return code [<return code>], error message : <error message>`	errorRole	BW-Plug-in ERROR_TRANSLATOR	This error points to a TIBCO Foresight® Translator return code. For example: `TIBCO-BW-PALETTE-HL7-501004:return code [158], error message : Error 44 : Access map failed` refers to Translator return code 158 "Could not open the map file."	Using the return code provided in the message, refer to the Return Codes section of the TIBCO Foresight® Translator documentation.
TIBCO-BW-PALETTE-HL7-501005	errorRole	BW-Plug-in ERROR_TRANSLATOR	An exception occurred during translation.	This is an internal product error. Contact TIBCO Support.

Error Code and Message	Role	Category	Description	Resolution
TIBCO-BW-PALETTE-HL7-501006	errorRole	BW-Plug-in ERROR_TRANSLATOR	An exception occurred during translation.	This is an internal product error. Contact TIBCO Support.
TIBCO-BW-PALETTE-HL7-501007 `Callback is checked, but no class selected.`	errorRole	BW-Plug-in ERROR_TRANSLATOR	The Callback checkbox is selected, but no class was specified.	Specify a Callback class name.
TIBCO-BW-PALETTE-HL7-502000 `The input data is required.`	errorRole	BW-Plug-in ERROR_INSTREAM	Input data is required.	Specify input data to be used for validation.
TIBCO-BW-PALETTE-HL7-502001 `Configuration error [{0}]`	errorRole	BW-Plug-in ERROR_INSTREAM	A configuration error has occurred.	Refer to the information in [{0}] to resolve the issue. Example: Configuration error **[missing guideline]**
TIBCO-BW-PALETTE-HL7-502003 `TIBCO-BW-PALETTE-HL7-502003: instream return code [<return code>], error message : <error message>`	errorRole	BW-Plug-in ERROR_INSTREAM	This error points to a TIBCO Foresight® Instream® return code. For example: `TIBCO-BW-PALETTE-HL7-502003: instream return code [140], error message : Access(R) failed` refers to Instream return code 140 "A critical error prevented validation from running successfully."	Using the return code provided in the message, refer to the Return Codes section of the TIBCO Foresight® Instream® documentation.

Error Code and Message	Role	Category	Description	Resolution
TIBCO-BW-PALETTE-HL7-502004 `Callback is checked, but no class selected.`	errorRole	BW-Plug-in ERROR_INSTREAM	The Callback checkbox is selected, but no class was specified.	Specify a Callback class name.
TIBCO-BW-PALETTE-HL7-502005 `Callback: Guideline name is missing.`	errorRole	BW-Plug-in ERROR_INSTREAM	A guideline is required for validation.	Specify a guideline to be used for validation.
TIBCO-BW-PALETTE-HL7-503000 `The input data is required.`	errorRole	BW-Plug-in ERROR_PARSEHL7 HEADER	Input data is required.	Specify input data to be used for header parsing.
TIBCO-BW-PALETTE-HL7-503001 `The input data encoding is not supported.`	errorRole	BW-Plug-in ERROR_PARSEHL7 HEADER	The specified input data encoding option is not supported.	Select a valid encoding option when defining parameters for the TranslateHL7 activity.
TIBCO-BW-PALETTE-HL7-503003 `MSH only contains [{0}] fields.`	errorRole	BW-Plug-in ERROR_PARSEHL7 HEADER	The specified HL7 document is invalid. The HL7 Message Header Segment does not contain the required number of fields.	Specify a valid HL7 document.
TIBCO-BW-PALETTE-HL7-503004 `Message Type field is missing.`	errorRole	BW-Plug-in ERROR_PARSEHL7 HEADER	The specified HL7 document is invalid. The HL7 document does not contain a Message Type field (required).	Specify a valid HL7 document.
TIBCO-BW-PALETTE-HL7-503005 `Invalid HL7 document.`	errorRole	BW-Plug-in ERROR_PARSEHL7 HEADER	The specified HL7 document is invalid.	Specify a valid HL7 document.

Error Code and Message	Role	Category	Description	Resolution
TIBCO-BW-PALETTE-HL7-503006 `Version ID is missing.`	errorRole	BW-Plug-in ERROR_PARSEHL7 HEADER	The specified HL7 document is invalid. The HL7 document does not have a Version ID (required).	Specify a valid HL7 document.
TIBCO-BW-PALETTE-HL7-504001 `The msaAckCode is incorrect, valid msaAckCode is as below: AA,AE,AR,A,E,R,accept,error, reject or CA,CE,CR.`	errorRole	BW-Plug-in ERROR_HL7 GENACK_INVALID_ACK_CODE	The msaAckCode is incorrect.	Correct the msaAckCode. Select a valid msaAckCode when defining parameters for the GenerateACK activity.
TIBCO-BW-PALETTE-HL7-504002 `The input data is required.`	errorRole	BW-Plug-in ERROR_HL7 GENACK_INVALID_ACK_CODE	Input data is required.	Specify input data to be used for acknowledgement.
TIBCO-BW-PALETTE-HL7- 505003 `INVALID PORT: Invalid Port; Port must be numeric (or separated by semicolon when it has more than one port). Error: [{0}]`	errorRole	BW-Plug-in EX_INVALID_PORT	The specified port is invalid. A port must be numeric. If more than one port is specified, they must be separated by a semicolon.	Specify a valid port number.
TIBCO-BW-PALETTE-HL7-505021 `EX_READ_TIMEOUT=Timeout while reading data`	errorRole	BW-Plug-in EX_READ_TIMEOUT	The process timed out while reading data.	Possible causes for this error are: (1) The data is being read before it is written. (2) The data being written is text/binary and it is being read as binary/text. (3) The data is written with/without separators and is being read without/with separators.

Error Code and Message	Role	Category	Description	Resolution
TIBCO-BW-PALETTE-HL7-505022 `IOException occurred while retrieving XML Output for activity [{0}].`	errorRole	BW-Plug-in EXCEPTION_OCCURED_RETRIEVE_RESULT	An IOException occurred while retrieving XML Output for the referenced activity.	This is an internal product error. Contact TIBCO Support.
TIBCO-BW-PALETTE-HL7-505023 `The LLPReceiver Process Starter activity [{0}] failed to a generate a new event due to exception [{1}] since it is in the stopped state.`	errorRole	BW-Plug-in TCP_RECEIVER_ILLEGAL_ARGUMENT_EXCEPTION	An exception occurred during the LLPReceiver Process Starter activity.	This is an internal product error. Contact TIBCO Support.
TIBCO-BW-PALETTE-HL7-505024 `The Wait For LLPReceiver activity [{0}] failed to a generate a new event due to exception [{1}] since it is in the stopped state.`	errorRole	BW-Plug-in WAIT_FOR_TCP_ILLEGAL_ARGUMENT_EXCEPTION	An exception occurred during the Wait For LLPReceiver activity.	This is an internal product error. Contact TIBCO Support.
TIBCO-BW-PALETTE-HL7-505025 `LLPReceiver : Socket exception [{0}]`	errorRole	BW-Plug-in EXCEPTION_OCCURED_LLPRECIVER_SOCK	An exception occurred during the LLPReceiver activity.	This is an internal product error. Contact TIBCO Support.

Error Code and Message	Role	Category	Description	Resolution
TIBCO-BW-PALETTE-HL7-505026 `Cannot start LLPReceiver on Host [{0}], Port [{1}]. Reason: [{2}].`	errorRole	BW-Plug-in EX_CANNOT_START_RECEIVER	The specified LLPReceiver cannot be started for the reason displayed in the error. Example: `Cannot start LLPReceiver on Host [localhost], Port [8999]. Reason: [java.net.BindException: Address already in use: JVM_Bind]`	Correct the host/port and retry.
TIBCO-BW-PALETTE-HL7-505027 `Separator is configured incorrectly {0}.`	errorRole	BW-Plug-in ERROR_RECEIVER_SEPARATOR_INVALID	The separator is configured incorrectly.	Correct separator and retry.
TIBCO-BW-PALETTE-HL7-505028: `Read timed out.`	errorRole	BW-Plug-in ERROR_RECEIVER_READ_DATA	The action timed out before completion.	Check the read timeout setting and adjust if necessary.
TIBCO-BW-PALETTE-HL7-506001 `LLPRESPONSE: Exception occurred while reply message for LLPResponse activity [{0}].`	errorRole	BW-Plug-in EXCEPTION_OCCURED_LLPRESPONSE_REPLYMESSAGE	An exception occurred during the LLPResponse activity.	This is an internal product error. Contact TIBCO Support.
TIBCO-BW-PALETTE-HL7-506002 `EX_ERROR_WRITING_DATA_GENERIC= Error writing data. Reason: [{0}], please check separator setting.`	errorRole	EXBW-Plug-in EX_ERROR_WRITING_DATA	An error occurred while writing the data.	Correct separator and retry.

Error Code and Message	Role	Category	Description	Resolution
TIBCO-BW-PALETTE-HL7-507001 `LLPRequestResponse: No existing socket found. host [{0}], port [{1}].`	errorRole	BW-Plug-in ERROR_LLPREQUEST_RESPONE_SOCKET_NOT FOUND	No existing socket was found for the specified host/ port.	Verify the server is running on the same port.
TIBCO-BW-PALETTE-HL7-507002 `LLPRequestResponse : Socket exception [{0}].`	errorRole	BW-Plug-in EXCEPTION_OCCURED_LLPREQUEST_RESPONE_SOCK	A socket exception occurred.	Refer to the information in [{0}] to resolve the issue. For example: `Socket exception [port out of range:89833].`
TIBCO-BW-PALETTE-HL7-507003: `LLPRequestResponse : time out waiting for Response.`	errorRole	BW-Plug-in ERROR_LLPREQUEST_RESPONE_TIMEOUT_FOR_RESPONE	The LLPRequestResponse activity timed out waiting for a response.	Set the LLPRequestResponse activity property "Request Timeout (msec)" to 0 (no time out).

HL7 Plug-in Example Processes

The TIBCO ActiveMatrix BusinessWorks™ Plug-in for HL7 examples are packaged with the installation of the plug-in.

This information describes how to run the HL7 Plug-in examples on a Microsoft Windows platform.

By default, the following examples are located in the <*BW Home*> \palettes\hl7*x.x*\samples directory:

- LLP Example - how to transport HL7 messages using Lower Level Protocol (LLP).
- Parser and Renderer Example - how to translate input data from one data format to another data format.
- Validate and Translate Example - how to validate input data and translate it from one data format to another data format using different methods of specifying validation guidelines and translation maps.

Setting Up and Running the Examples

Use the following information to set up and run TIBCO ActiveMatrix™ BusinessWorks Plug-in for HL7 examples.

You should be familiar with the plug-in before running the examples.

Copy Guidelines and Maps

The HL7 Plug-in examples make use of guidelines and maps. Before using any of the examples, you must copy the associated files to the appropriate directory.

For each example, **guidelines** and **map** files are found in the example's \Guidelines directory. For example: `\bw\palettes\hl7\x.x\samples\ParserAndRenderer\DataAndGuidelines\Guidelines`.

Copy the guideline and map files as follows:

1. Copy the .std, .xsd, and .map files to Translator's \Database directory.
2. Copy the .std files to Instream's \Database directory.

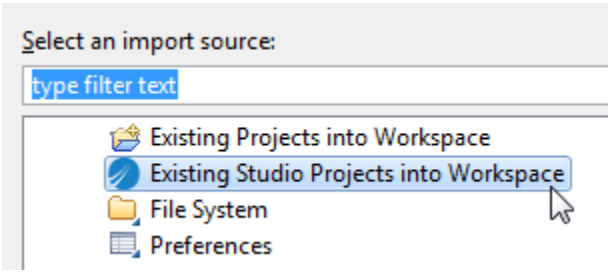
(Note that the .std files are copied to the \Database directory for BOTH TIBCO Foresight® Instream® and TIBCO Foresight® Translator.)

Importing an Example

Use the following steps to import an example.

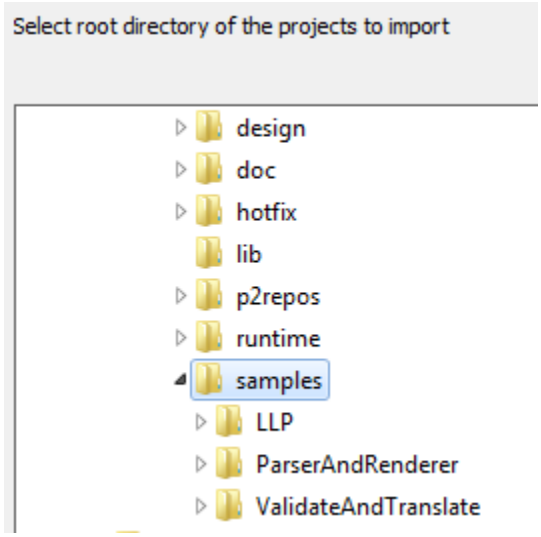
1. Select **File > Import**.

   The Import screen appears.
2. Select Existing Studio Projects into Workspace.



The Browse For Folder screen appears.

3. Browse to the location of the desired HL7 samples. By default this is the <*BW_Home*> \palettes \hl7*x.x*\samples directory, as shown here:



The path will be different if you have saved the samples to another location.

Optional: Set Environment Variables for the Process

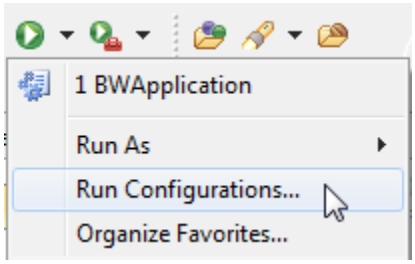
Instream and Translator environment variables should have been set as part of the Plug-in for HL7 Post-installation tasks (refer to *TIBCO ActiveMatrix BusinessWorks™ Plug-in for HL7 Installation*). Doing so tells ActiveMatrix BusinessWorks where to access the Instream and Translator executables when the products are used in a TIBCO ActiveMatrix™ BusinessWorks HL7 Plug-in process.

It is also possible to specify environment variables on a per-process basis. The following procedure instructs the process to utilize the variables specified for the process instead of those specified for TIBCO ActiveMatrix™ BusinessWorks at the higher level.

Refer to the appropriate section for your operating system:

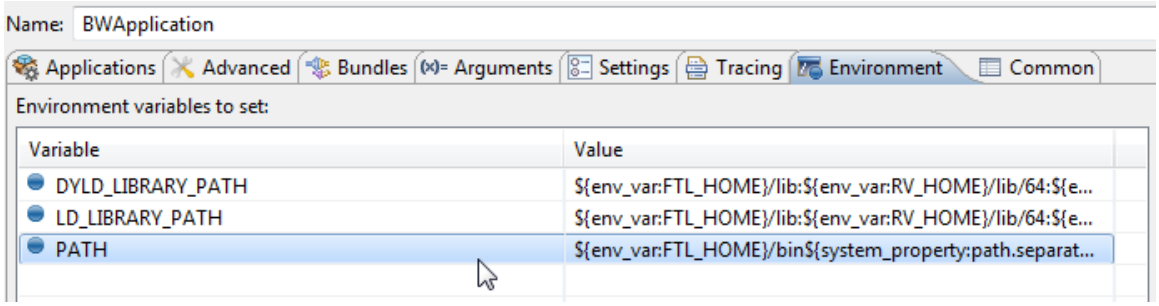
For Windows Platforms

1. Open the desired process and click within the process.
2. Select **Run > Run Configurations...**.



The Run Configuration screen appears.

3. Select the environment tab and double click on PATH.



The Edit Environment Variable screen appears.

4. Add the path to the desired location of the Instream\bin and Translator\bin directories at the beginning of the string, before the first $ character, and with a **;** at the end of each segment of added text.

   Do NOT delete the existing string, simply add information to the beginning.

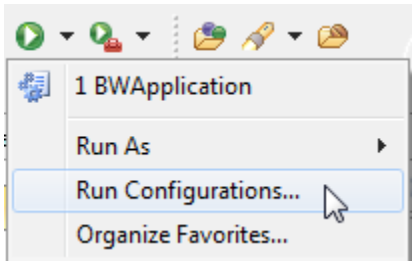
   Example:

   C:\tibco64\instream\8.5\bin**;**C:\tibco64\translator\3.5\bin**;**${env_var:FTL_HOME}.....

5. Select the Apply button.
6. Select the Close button.

For Unix Platforms

1. Open the desired process and click within the process.
2. Select **Run > Run Configurations...**.



The Run Configuration screen appears.

3. Select the environment tab and double click on LD_LIBRARY_PATH.

   The Edit Environment Variable screen appears.

4. Add the path to the desired location of the Instream/bin and Translator/bin directories at the beginning of the string, before the first $ character, and with a **:** at the end of each segment of added text.

   Do NOT delete the existing string, simply add information to the beginning.

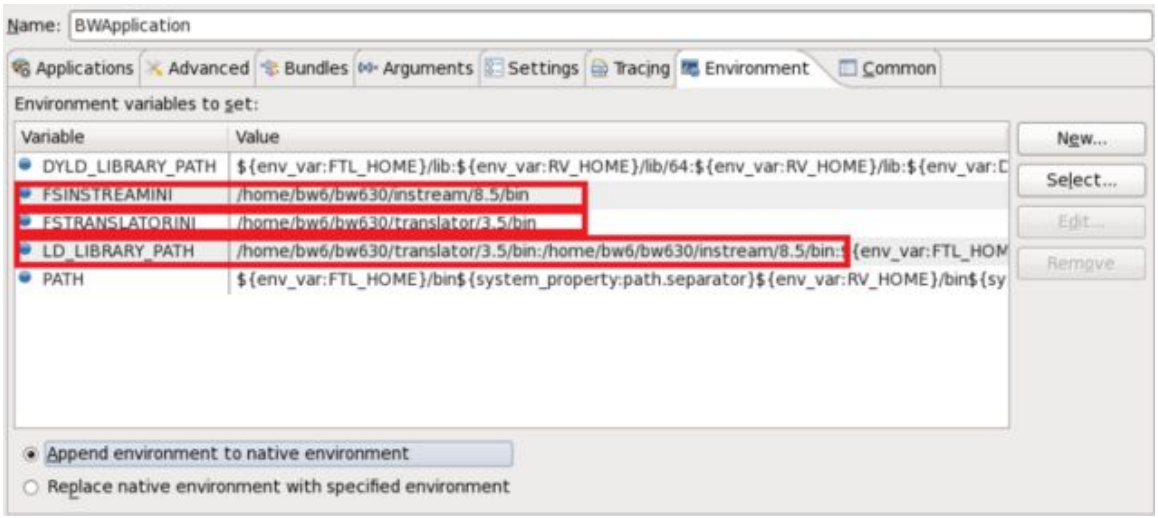
   Example

   /home/bw/bw630/instream/8.5/bin**:**/home/bw/bw630/translator/3.5/bin**:**${env_var:FTL_HOME}.....

5. Select the Apply button.
6. Select the New button.
7. Add a new variable FSINSTREAMINI and specify the desired location of the Instream\bin directory.

8. Select the Apply button.
9. Select the New button.
10. Add a new variable FSTRANSLATORINI, and specify the desired location of the Translator\bin directory.
11. Select the Apply button.
12. Select the Close button.

This example shows a sample Run Configuration screen after the variables have been set for Unix.

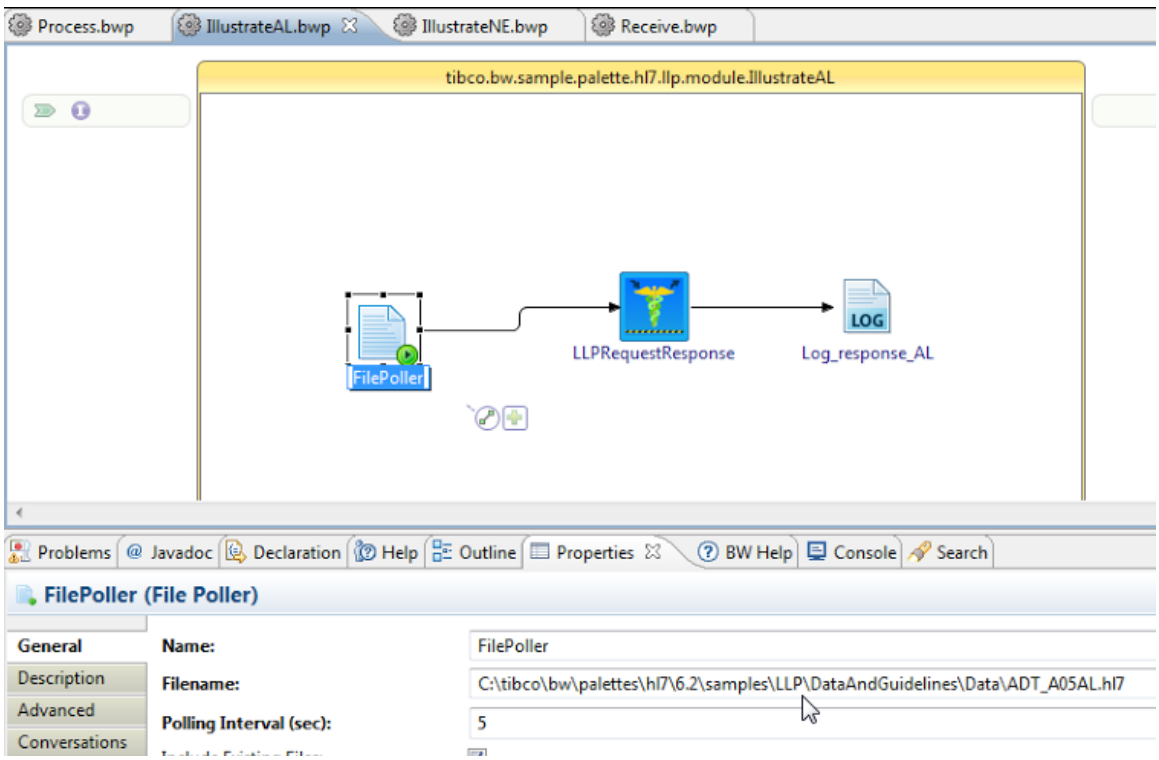


Verify File Paths in the Examples

Before running an example, verify the file path used in the example is correct for your environment.

1. Open the desired example.

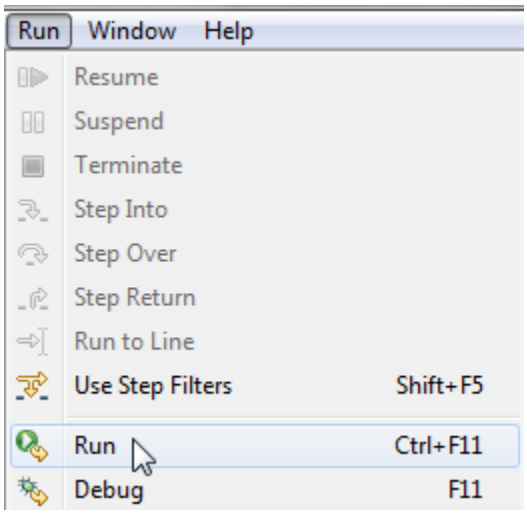
   In this image, the IllustrateAL.bwp process is opened.



2. Check the file path. In this example the FilePoller activity is set to pick up the input file at the following path <*BW_HOME*>\samples\palette\hl7*x.x*\samples\LLP\DataAndGuidelines\Data\ADT_A05AL.hl7. You may need to alter the file path if you have saved the samples to a different location.

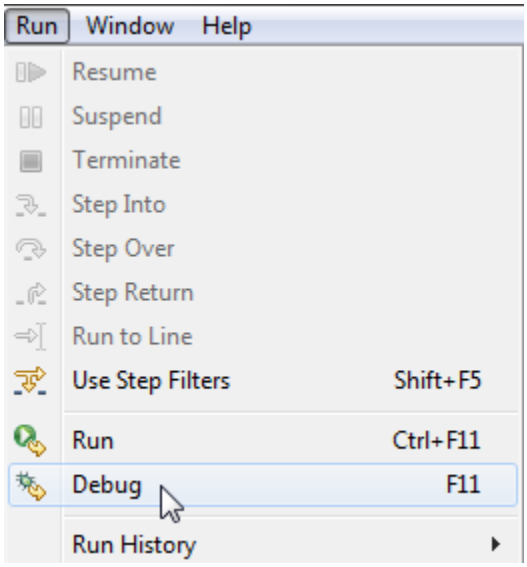
Running an Example

1. Open the desired example.
2. Click **Run > Run** to run the application.



After the process executes successfully, a Success message is written to the Console. Select **Window > Show Console** in TIBCO Business Studio™ to view the log messages.

3. Optional: Click **Run > Debug** to debug the application.



The perspective changes to Debug from Modeling. You can view the job details in the Console view from the Debug perspective.

LLP Example

The `\bw\palettes\hl7\x.x\samples\LLP` folder contains sample projects that help you understand how to transport HL7 messages using Lower Level Protocol (LLP).

For more information about the LLP activities, see LLPReceiver, LLPRequestResponse, and LLPResponse.

Example Data Files

These examples make use of the data files found in the `\bw\palettes\hl7\x.x\samples\LLP \DataAndGuidelines\Data` folder:

- ADT_A05NE.hl7 - The HL7 message file that triggers the Illustrate NE process.
- ADT_A05AL.hl7 - The HL7 message file that triggers the Illustrate AL process.

Process Description

This example contains the following predefined processes:

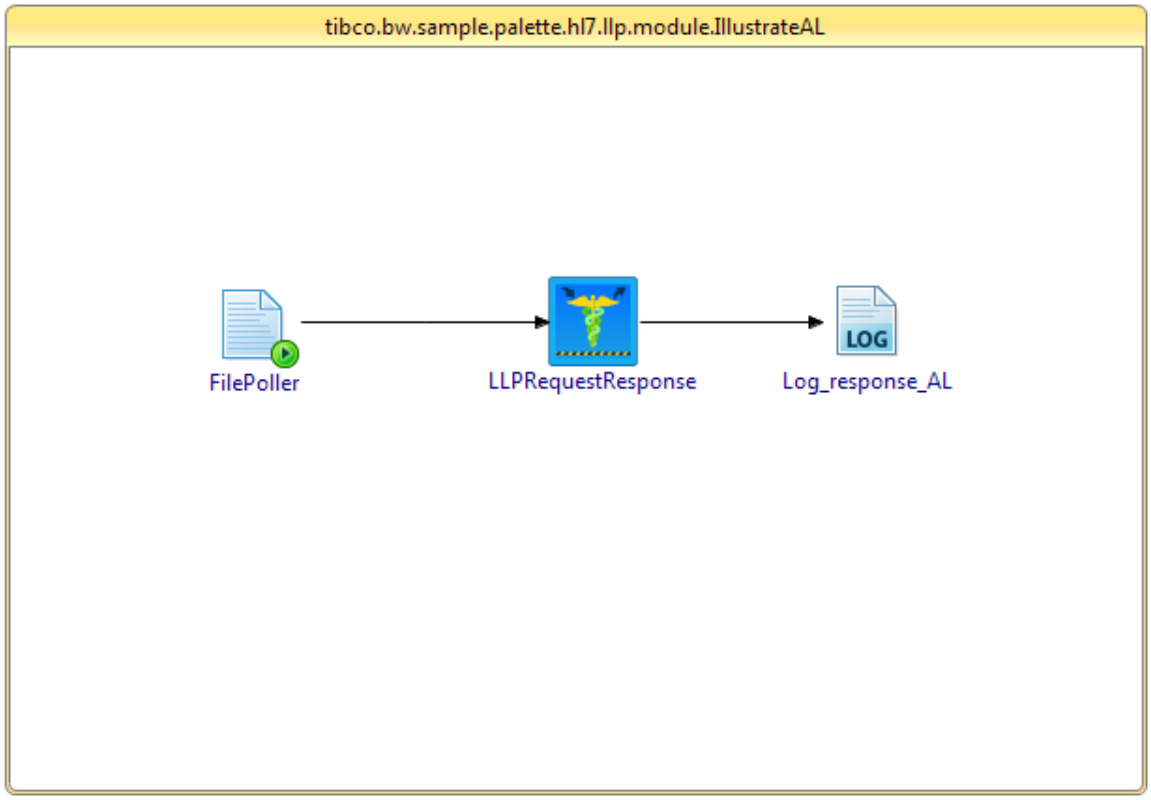
- Illustrate AL Process
- Illustrate NE Process
- Receive Process

Illustrate AL Process

The Illustrate AL process sends the HL7 message string to the server and receives an acknowledgement message in response.

Example Process

The Illustrate AL Process is illustrated here:



Process Definition

The process performs the following operations:

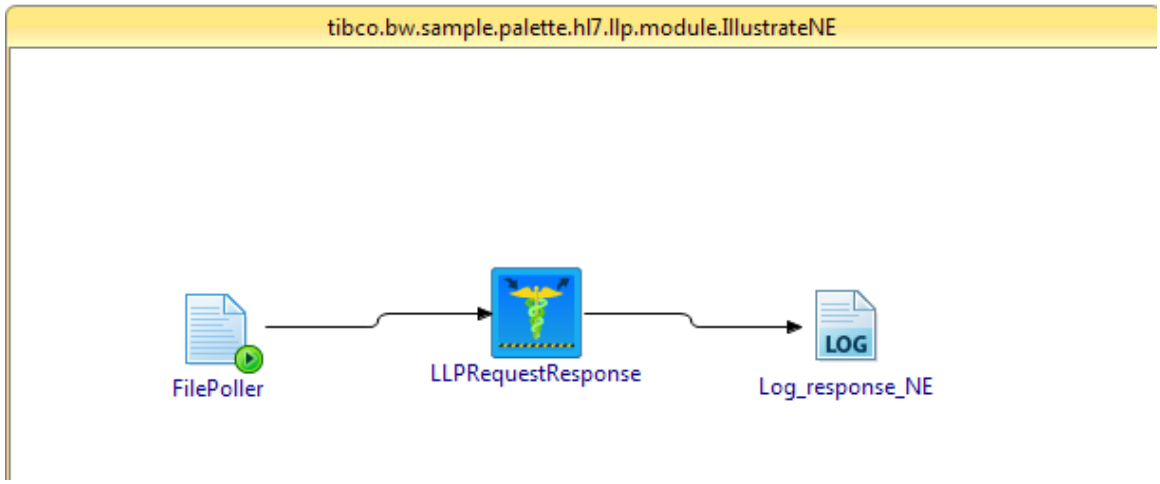
1. The FilePoller activity polls the file named ADT_A05AL.hl7 in the directory specified in the File Name field of the Configuration tab. If the file exists then the process starts.
2. The LLP Request Response activity sends the HL7 messages in the ADT_A05AL.hl7 file to the server, and then receives an acknowledgement message as response from the server.
3. After the process executes successfully, a Success message is written to the Console. Select **Window > Show Console** in TIBCO Business Studio™ to view the log messages.

Illustrate NE Example

The Illustrate NE process sends the HL7 message string to the server and expects no response.

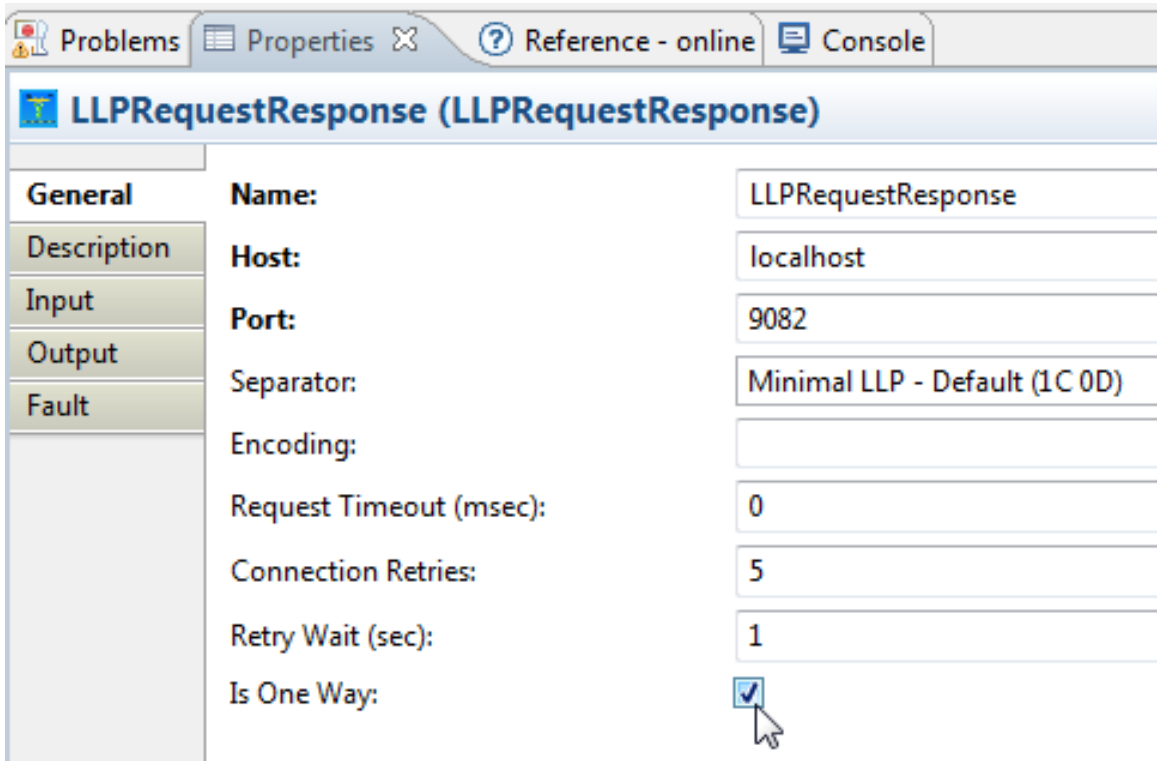
Example Process

The Illustrate NE Process is illustrated here:



Process Definition

This process is similar to the Illustrate AL process with one exception: it expects no response. This option is configured via the LLPRequestResponse activity General properties tab using the "Is One Way" checkbox, as shown here.



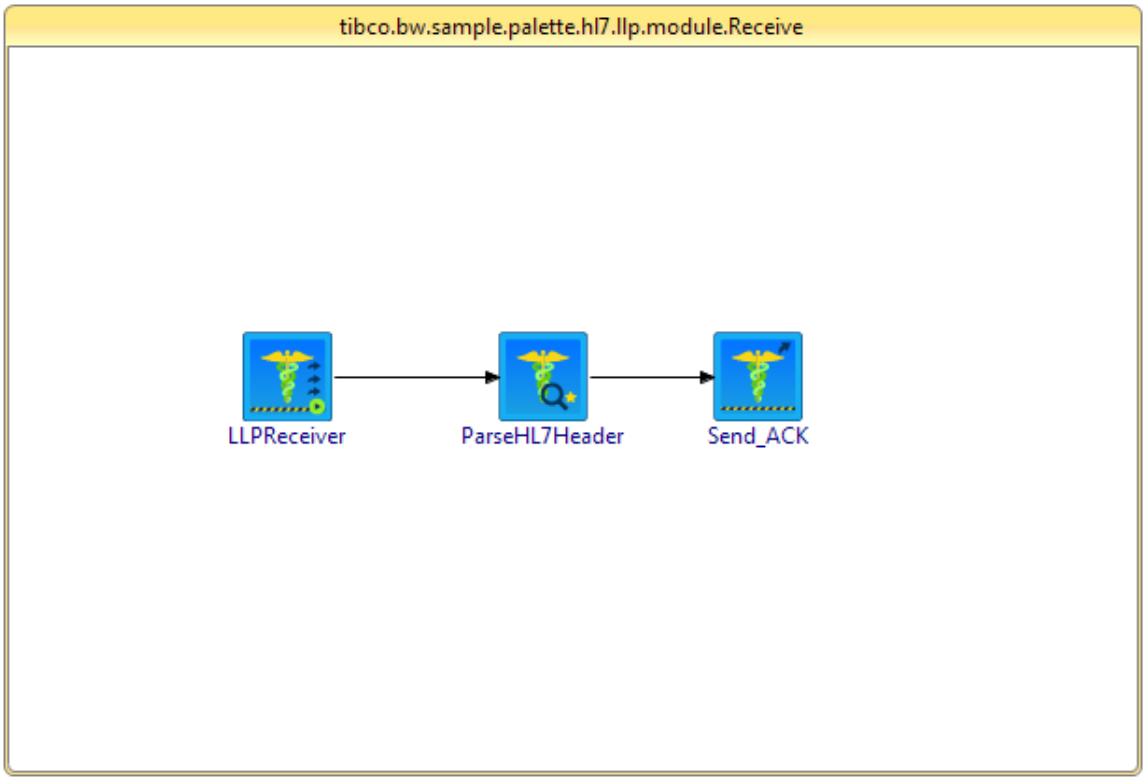
See Illustrate AL Process for more details on the Process Definition.

Receive Example

The Receive process parses the header of the received HL7 messages and sends the generated acknowledgement response back depending on the acknowledgement type.

Example Process

The Receive Process is illustrated here:



Process Definition

The process performs the following operations:

1. The HL7 LLP Receiver activity receives the HL7 messages sent from the client. The process starts.
2. The Parse HL7 Header activity parses the header of the received HL7 messages and generates its acknowledgement.
3. After the process executes successfully, a Success message is written to the Console. Select **Window > Show Console** in TIBCO Business Studio™ to view the log messages.

Parser and Renderer Example

The `\bw\palettes\hl7\x.x\samples\ParserAndRenderer` folder contains sample projects that help you understand how to translate input data from one data format to another data format.

For more information about translating data, see TranslateHL7.

Example Data Files

These examples make use of the following data file found in the `\bw\palettes\hl7\x.x\samples\ParserAndRenderer\DataAndGuidelines\Data` area:

- ADT_A05.hl7 - The HL7 ADT_A05 message that triggers the ParseAndRender process.

Example Guideline and Map Files

These examples make use of the following guideline and map files found in the `\bw\palettes\hl7\x.x\samples\ParserAndRenderer\SchemaFiles\Guidelines` area:

- HL7_26_ADT_A05.std - The schema definition for an ADT_A05 message in HL7 format. This is the source guideline that describes the message before translation.

- HL7_26_ADT_A05.xsd - The guideline for an ADT_A05 message in XML format. This is the target guideline that describes the message after translation.
- HL7_26_ADT_A05_EX.map - The predefined map file that is used to translate the ADT_A05 message from HL7 format into XML format.
- HL7_26_ADT_A05_XE.map - The predefined map file that is used to translate the ADT_A05 message from XML format into HL7 format.
- HL7_26_ADT_A28.std - The schema definition for an ADT_A28 message in HL7 format. This is the target guideline that describes the message after translation.
- HL7_26_ADT_A28.xsd - The guideline for an ADT_A28 message in XML format. This is the source guideline that describes the message before translation.
- HL7_26_ADT_A28_EX.map - The predefined map file that is used to translate the ADT_A28 HL7 format into XML format.
- HL7_26_ADT_A28_XE.map - The predefined map file that is used to translate the ADT_A28 message from XML format into HL7 format.

Process Description

This example contains the following predefined process:

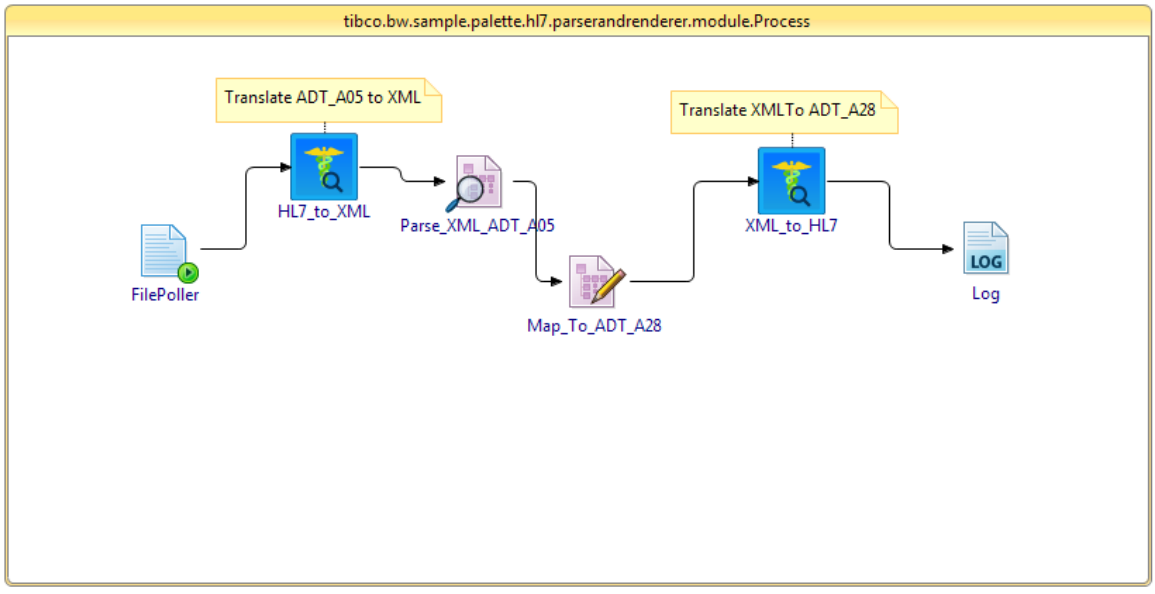
- Parse and Render Process

Parser and Renderer Process

The ParserAndRenderer example shows how to translate input data from one data format to another data format using parsing. In this case, the ADT_AO5.hl7 data is transformed into ADT_A28.hl7 data using the TranslateHL7 activity to map and translate data.

Example Process

The Parser and Renderer process is illustrated here:



Process Definition

The process performs the following operations:

1. The FilePoller activity polls the file named ADT_A05.hl7 in the directory specified in the File Name field of the Configuration tab. If the file exists then the process starts.

The ADT_A05.hl7 file is passed to the Translate_ADTA05_toXML activity.

2. The Parse ADT_A05 activity translates the ADT_A05.hl7 file into an XML string based on the HL7_26_ADT_A05_EX.map file.

   The XML string is passed to the Parse XML ADT_A05 activity.

3. The Parse XML ADT_A05 activity parses the input XML string into an XML schema tree based on the HL7_26_ADT_A05.xsd file that is specified in the Output Editor tab.

   The parsed XML schema is passed to the Map To ADT_A28 activity.

4. The Map To ADT_A28 activity takes the parsed XML schema and renders it as an XML string based on the HL7_26_ADT_A28.xsd file that is specified in the Input Editor tab.

   The XML string is passed to the XML To HL7 activity.

5. The XML To HL7 activity translates the input XML string into the HL7 message.

Validate and Translate Example

The `\bw\palettes\hl7\x.x\samples\ValidateAndTranslate` folder contains a sample project that helps you understand how to validate input data and translate it from one data format to another data format using different methods of specifying validation guidelines and translation maps.

For more information about validating and translating data, see ValidateHL7 and TranslateHL7.

Example Data Files

These examples make use of the following data file found in the `\bw\palettes\hl7\x.x\samples\ValidateAndTranslate\DataAndGuidelines\Data` area:

- VXR_V03_ascii.txt - The EDI text file that triggers the ValidateandTranslate process.

Example Guideline and Map Files

These examples make use of the following guideline and map files found in the `\bw\palettes\hl7\x.x\samples\ValidateAndTranslate\DataAndGuidelines\Guidelines` area:

- VXR_V03.std - The schema definition for an VXR_V03 message in ASCII text format. This is the source guideline that describes the message before translation.
- VXR_V03_VXR_V03.xsd - The guideline for an VXR_V03 message in XML format. This is the target guideline that describes the message after translation.
- VXR_V03_VXR_V03_EX.map - The predefined map file that is used to translate the VXR_V03 message from HL7 format into XML format.
- VXR_V03_VXR_V03_XE.map - The predefined map file that is used to translate the VXR_V03 message from XML format into HL7 format.

Process Description

This example contains the following predefined process:

- Validate and Translate Process

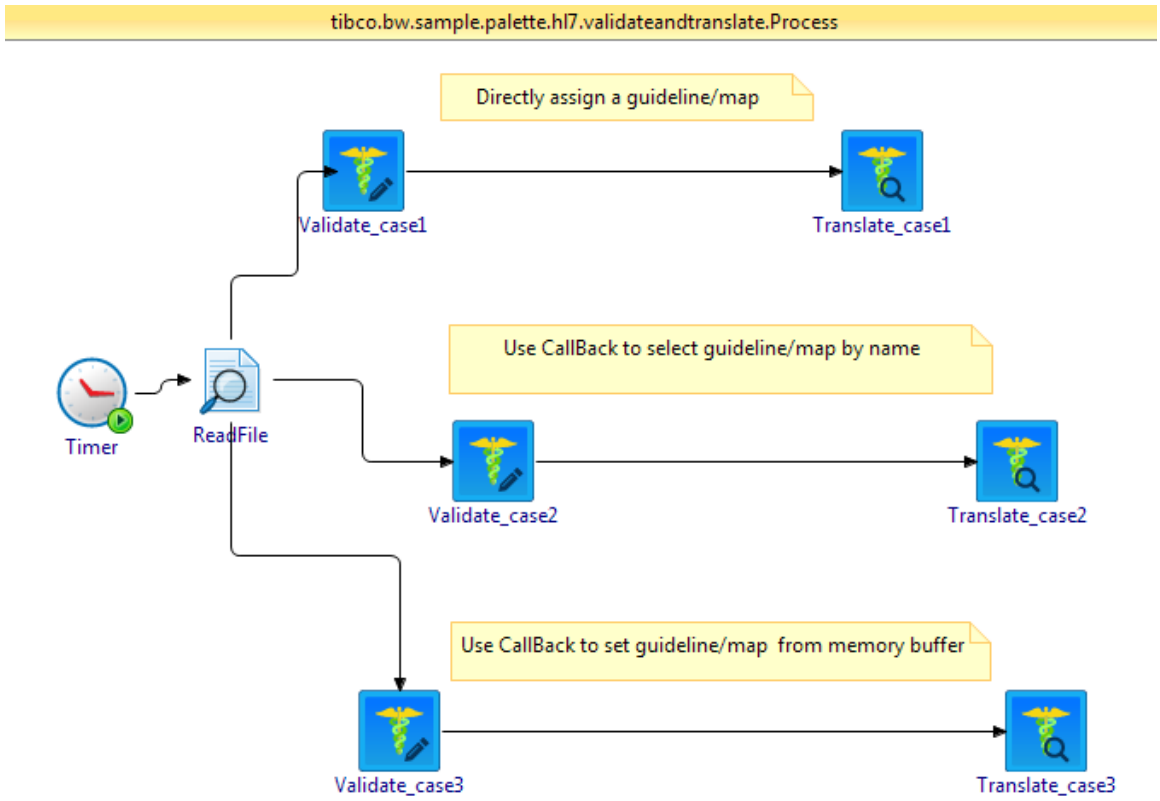
Validate and Translate Process

The ValidateAndTranslate example shows how to validate input data and translate it from one data format to another data format using three different methods of specifying validation guidelines and translation maps.

For more information about validating and translating data, see ValidateHL7 and TranslateHL7.

Example Process

The Validate and Translate process is illustrated here:



Process Definition

The process illustrates validation of an HL7 file and translation of the file to XML format using three different methods.

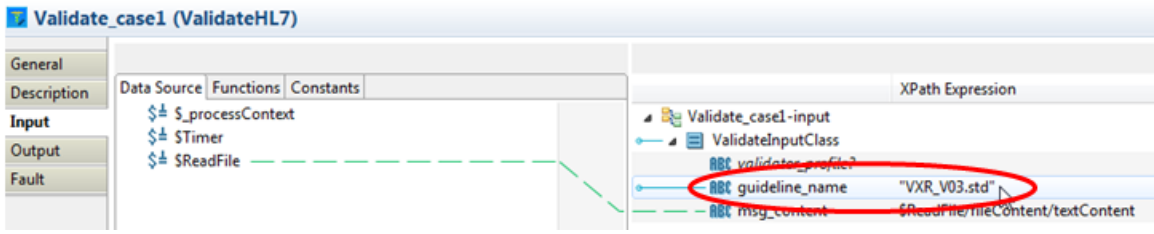
- Validation and Translation directly assigning a guideline and map
- Validation and Translation using a Callback to select a guideline and map by name
- Validation and Translation using a Callback to select a guideline and map from memory

Validation and Translation directly assigning a guideline and map

This process performs the following operations:

1. The FilePoller activity polls the file named VXR_V03_ascii.txt in the directory specified in the File Name field of the Configuration tab. If the file exists then the process starts.

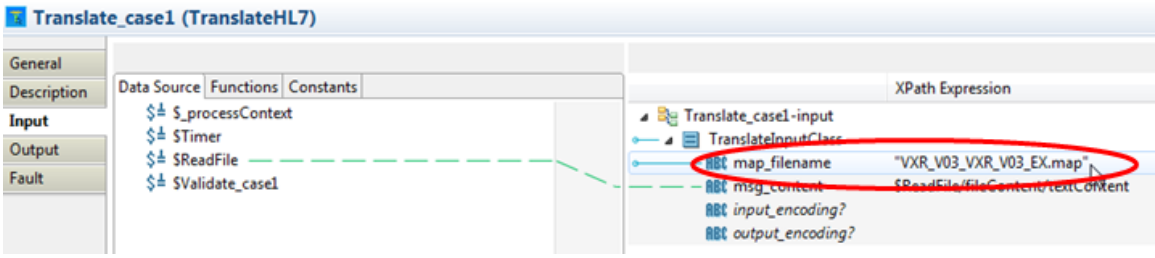
   The VXR_V03_ascii.txt file is passed to the Validate_case1 activity.
2. The Validate_case1 activity validates VXR_V03_ascii.txt file using the VXR_V03.std file. The VXR_V03.std file has been specified directly during configuration of the Validate_case1 activity, as shown here:



The validated file is passed to the Translate_case1 activity.

3. The Translate_case1 activity translates the validated file using the VXR_V03_VXR_V03_EX.map file.

   The VXR_V03_VXR_V03_EX.map file has been specified directly during configuration of the Translate_case1 activity, as shown here:



Translate_case1 activity translates the input HL7 string into an XML format.

For more information about validating and translating data, see ValidateHL7 and TranslateHL7.

Validation and Translation using a CallBack to select a guideline and map by name

This process performs the following operations:

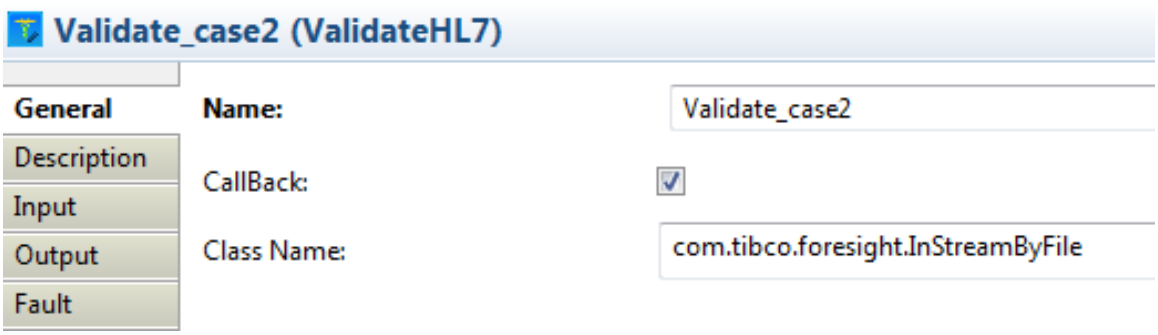
1. The FilePoller activity polls the file named VXR_V03_ascii.txt in the directory specified in the File Name field of the Configuration tab. If the file exists then the process starts.

   The VXR_V03_ascii.txt file is passed to the Validate_case2 activity.

2. The Validate_case2 activity validates VXR_V03_ascii.txt file using a guideline selected with a ValidateHL7 CallBack. ValidateHL7 CallBack allows you to select validation guidelines and validation profiles based on the contents of the input data by modifying Java code.

   In this case, the available guidelines exist **on disk** and the correct guideline is selected using a CallBack based on information contained in the input file header information. For example, if the input file header contains the value "X", the CallBack knows to select guideline "Y".

   For more information about validating data using CallBacks, see ValidateHL7 CallBack.

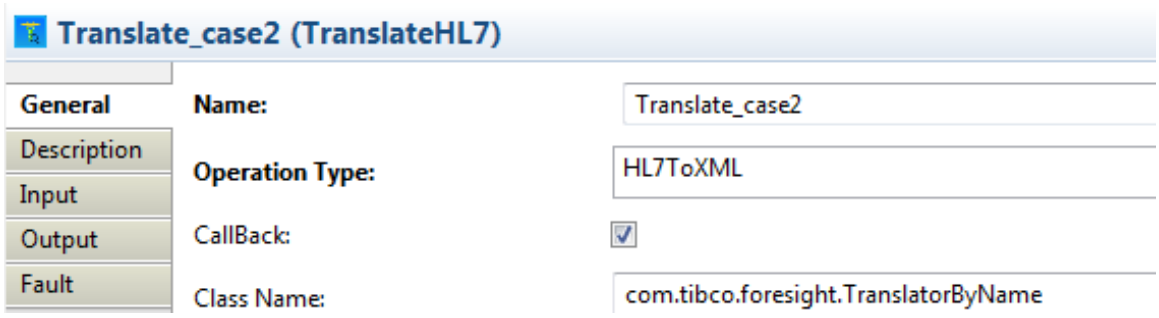


The validated file is passed to the Translate_case2 activity.

3. The Translate_case2 activity translates the validated file using a translation map specified with a TranslateHL7 CallBack. TranslateHL7 CallBack allows you to select a translation map based on the contents of the input data by modifying Java code.

In this case, the available translation maps exist **on disk** and the correct map is selected using a CallBack based on information contained in the input file header information. For example, if the input file header contains the value "X", the CallBack knows to select translation map "Y".

For more information about translating data using CallBacks, see TranslateHL7 CallBack.



Translate_case2 activity translates the input HL7 string into an XML format.

Validation and Translation using a Callback to select a guideline and map from memory

This process performs the following operations:

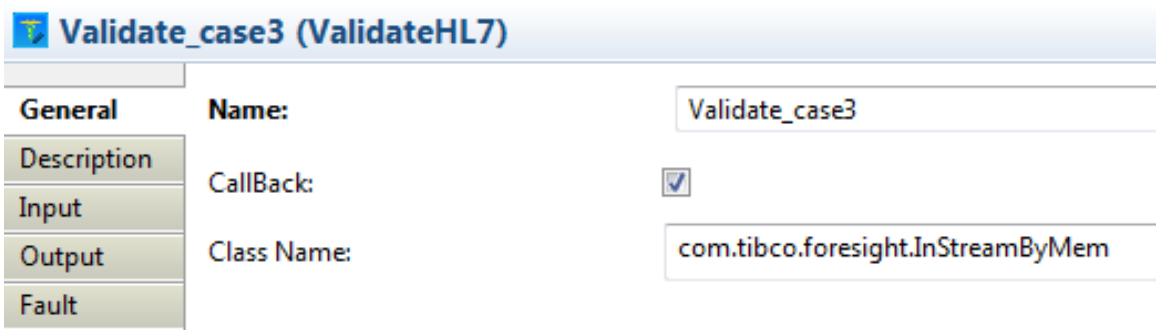
1. The FilePoller activity polls the file named VXR_V03_ascii.txt in the directory specified in the File Name field of the Configuration tab. If the file exists then the process starts.

   The VXR_V03_ascii.txt file is passed to the Validate_case3 activity.

2. The Validate_case3 activity validates VXR_V03_ascii.txt file using the a guideline specified with a ValidateHL7 CallBack. ValidateHL7 CallBack allows you to select validation guidelines and validation profiles based on the contents of the input data by modifying Java code.

   In this case, the available guidelines exist **in the memory buffer** and the correct guideline is selected using a CallBack based on information contained in the input file header information. For example, if the input file header contains the value "X", the CallBack knows to select guideline "Y".

   For more information about validating data using CallBacks, see ValidateHL7 CallBack.

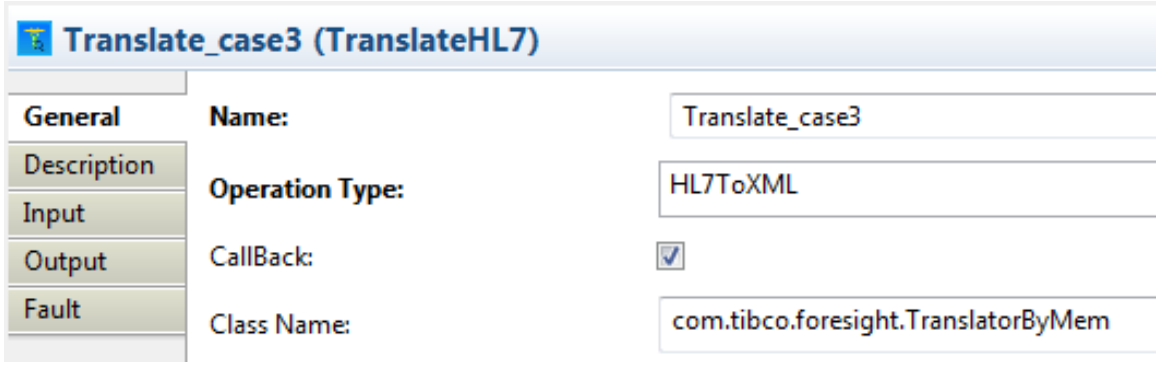


   The validated file is passed to the Translate_case3 activity.

3. The Translate_case3 activity translates the validated file using a translation map specified with a TranslateHL7 CallBack. TranslateHL7 CallBack allows you to select a translation map based on the contents of the input data by modifying Java code.

   In this case, the available translation maps exist **in the memory buffer** and the correct map is selected using a CallBack based on information contained in the input file header information. For example, if the input file header contains the value "X", the CallBack knows to select translation map "Y".

   For more information about translating data using CallBacks, see TranslateHL7 CallBack.



Translate_case3 activity translates the input HL7 string into an XML format.

Introduction to TIBCO Foresight EDISIM

TIBCO Foresight® EDISIM® provides an intuitive GUI interface that allows you to:

- create and customize EDI, XML, and Flat File guidelines
- check EDI data for compliance to a published standard or a guideline
- build translation maps.

The following general processes will aid you in working with validation guidelines and translation maps for use with the TIBCO ActiveMatrix BusinessWorks™ Plug-in for HL7:

- Creating XSD and Map Files
- Validating a Message
- Mapping Files

For complete information on how to use TIBCO Foresight® EDISIM®, see TIBCO Foresight® EDISIM® documentation.

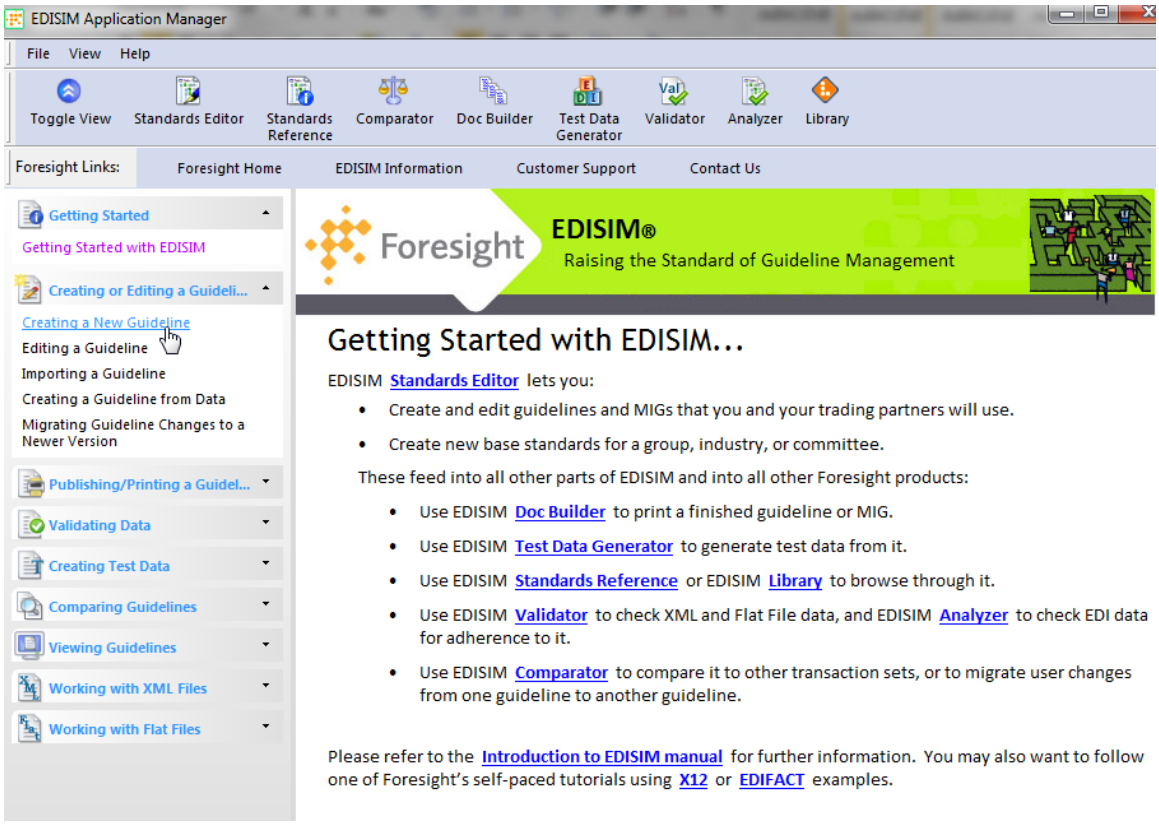
For complete information on how to use TIBCO Foresight® Translator, see TIBCO Foresight® Translator documentation.

Creating XSD and MAP files

The TIBCO Foresight® EDISIM® Standards Editor allows you to generate XSD and MAP files for translation. This section gives an example of how to use Standards Editor to generate an XSD file and a MAP file.

To generate XSD and MAP files, complete the following steps:

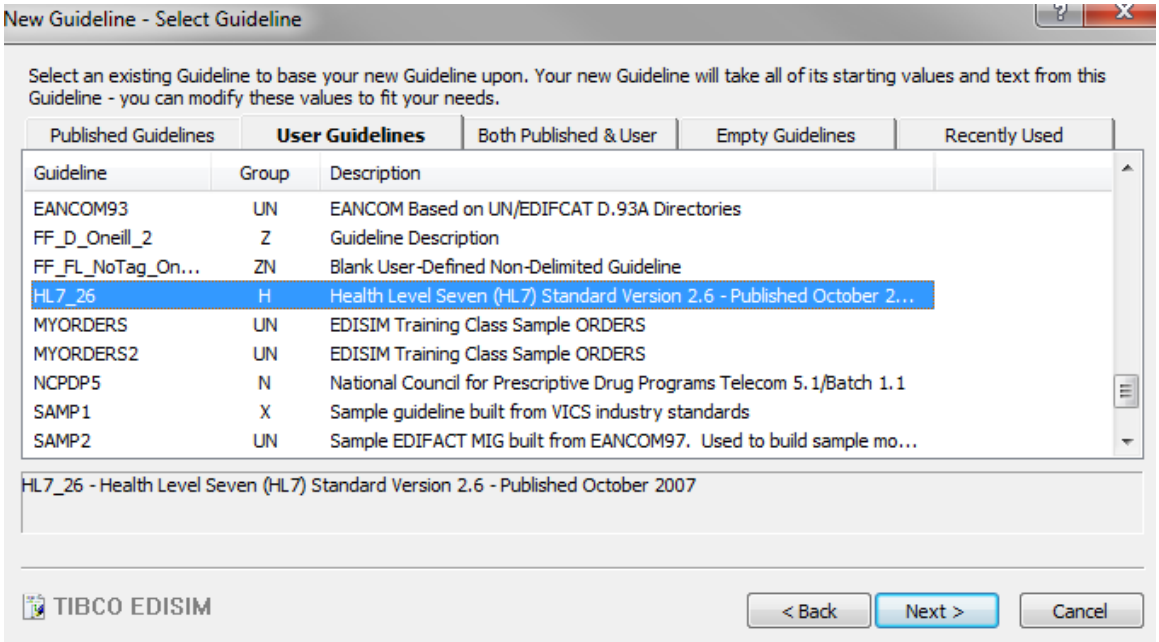
1. Start TIBCO Foresight® EDISIM® Application Manager. Select **Start > All Programs >** TIBCO_HOME **> EDISIM** in Windows. The EDISIM Application Manager window appears.
2. Create a guideline. Click the **Creating Or Editing A Guideline** item from the left-hand panel and then click the **Creating A New Guideline** item.



3. Click the **Standards Editor** link on the right panel. The Empty Guideline - EDISIM window appears.
4. Create a new standard. For example, create a new standard based on the HL7_26 guideline.

   a. Select **File > New** from the Empty Guideline - Edisim window. The New Guideline - Select Guideline dialog appears.

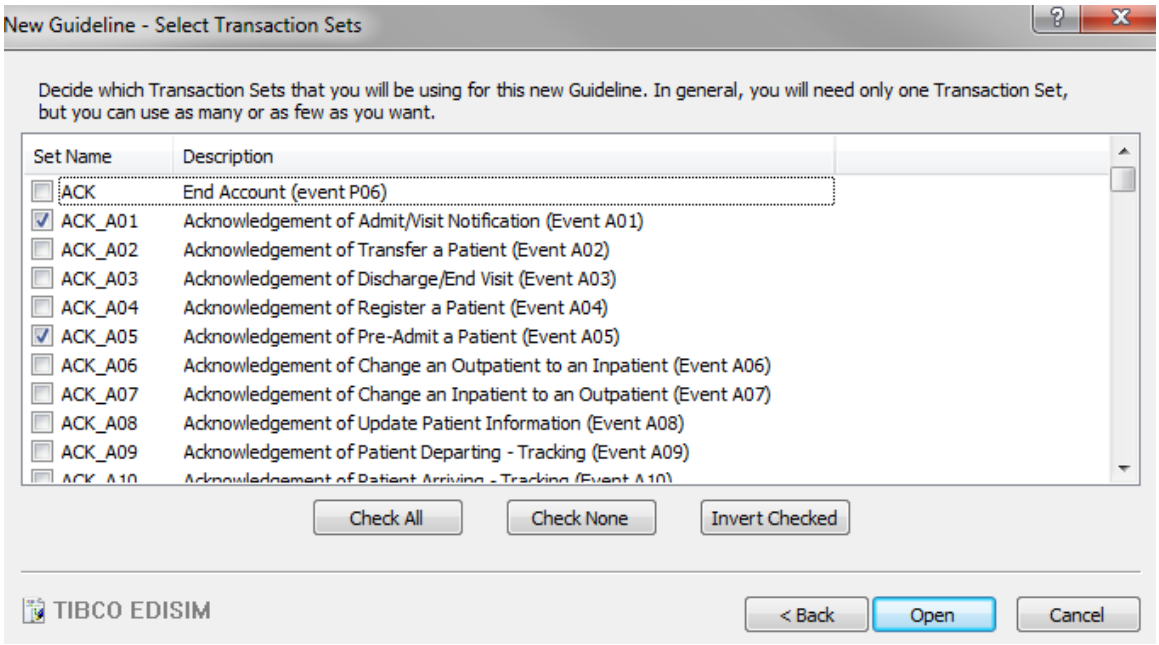
   b. Click the **User Guideline** tab. Select the guideline from the User Guideline tab. In this example, we select the **HL7_26** item.



   c. Click the **Next** button. The New Guideline - Select Transaction Sets dialog appears.

d. Select messages you want to include in your new guideline. For example, the ACK_A01 and ACK_A05 messages.

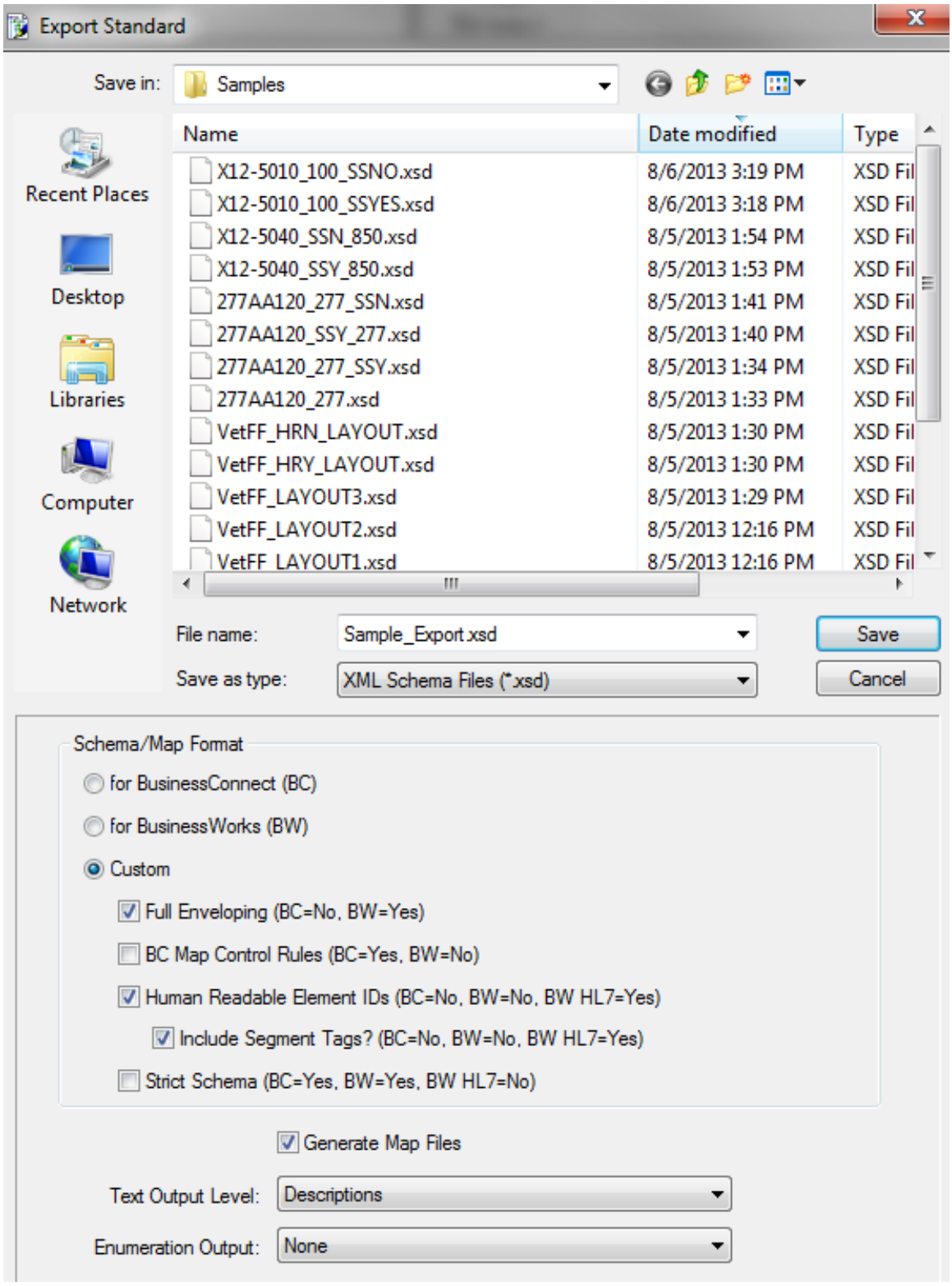
e. Click the **Open** button.



5. Edit the newly created guideline based on your requirements. For example, add Z segments, remove unused segments, or adjust optionality of fields in the guideline.
6. Save the guideline. Select **File > Save** in the menu. The Save Guideline As dialog appears. Type a guideline name in the Name field and then click the **Save** button.
7. Export the guideline as a schema.

   a. Select **File > Export > Export Current Guideline > To Schema** from the menu. The Export Standard dialog appears.

   To enable the Export Current Guide function, you must save the guideline first



b. Choose a path to save the guideline.

c. In the Schema/Map Format panel, select the **Custom** radio button and then check the **Full Enveloping (BC=No, BW=Yes)** , **Human Readable Element IDs (BC=No, BW=No, BW HL7=Yes)** , and **Include Segment Tags** checkboxes.

Uncheck ***Strict Schema*** **(BC=Yes, BW=Yes, BW HL7=No)**

d. Check ***Generate Map Files***.

e. Click the **Save** button to save the file. An XSD file and a MAP file are generated.

For more information on how to use TIBCO Foresight® EDISIM® modules, see TIBCO Foresight® EDISIM® documentation.

Validating a Message

TIBCO Foresight® EDISIM® Validator checks the validity of data based on the standards and guidelines.

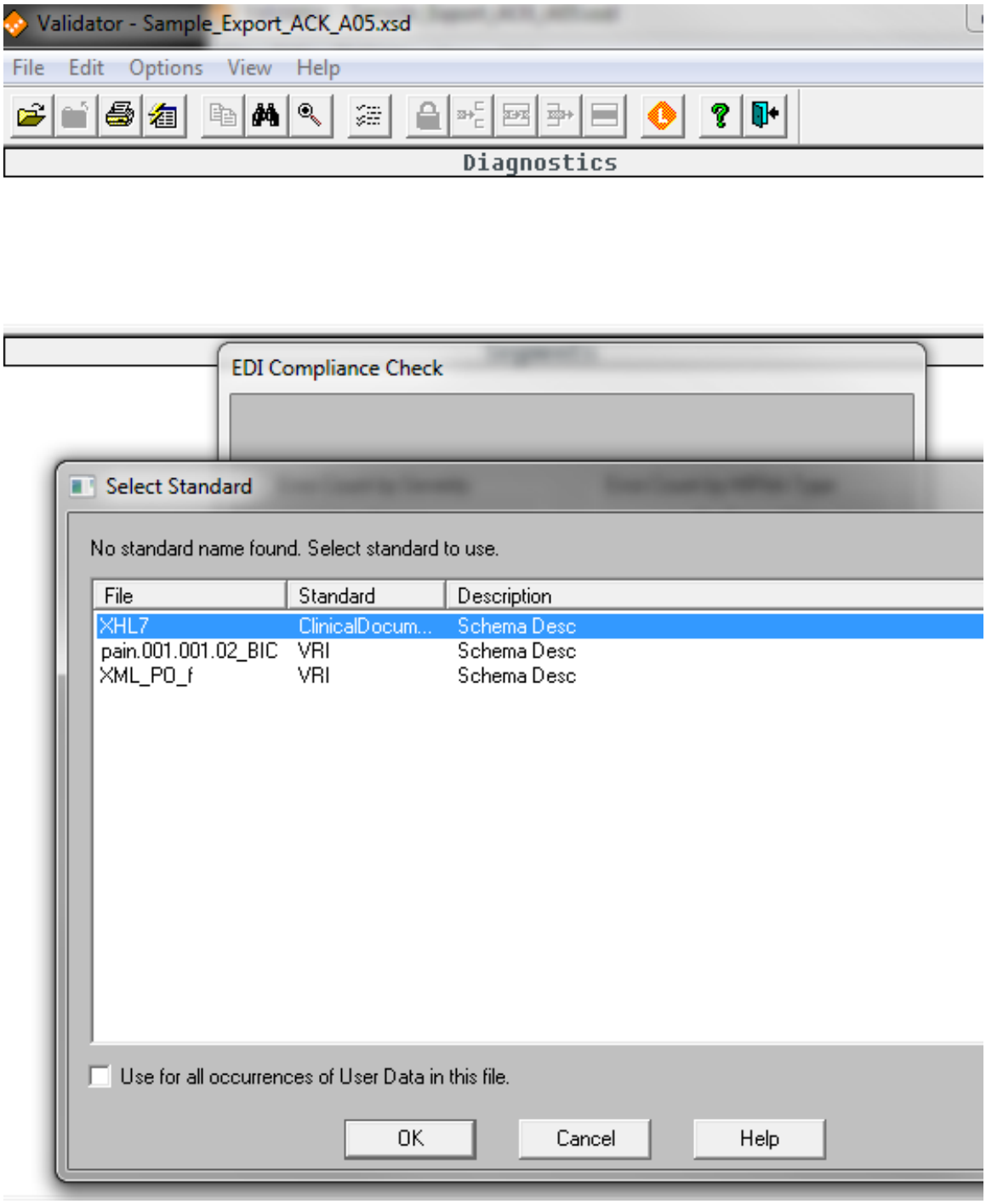
For complete information about how to use Validator, see TIBCO Foresight® EDISIM® documentation.

To validate messages, follow these steps:

1. Open TIBCO Foresight® EDISIM® Validator. You can do so in one of the following ways:

   - Select **Start > All Programs >** TIBCO_HOME **> EDISIM > Validator** in Windows.

   - Click the  button in the TIBCO Foresight® EDISIM® Standards Editor toolbar.
2. Select **File > Open** in the Validator menu to open an HL7 message instance. The Select Standard dialog appears.
3. Choose a guideline to verify the message. Select a standard from the Select Standard dialog and then click the **OK** button to start checking the message. For example, select the XHL7 file as a standard.



4. When the validation is finished, click the **OK** button to dismiss the Analysis Completed box.

Creating Mapping Files

Use TIBCO Foresight® Translator to create mapping files.

For complete information on how to create mapping files, see TIBCO Foresight® Translator documentation.

To create mapping files, follow these steps:

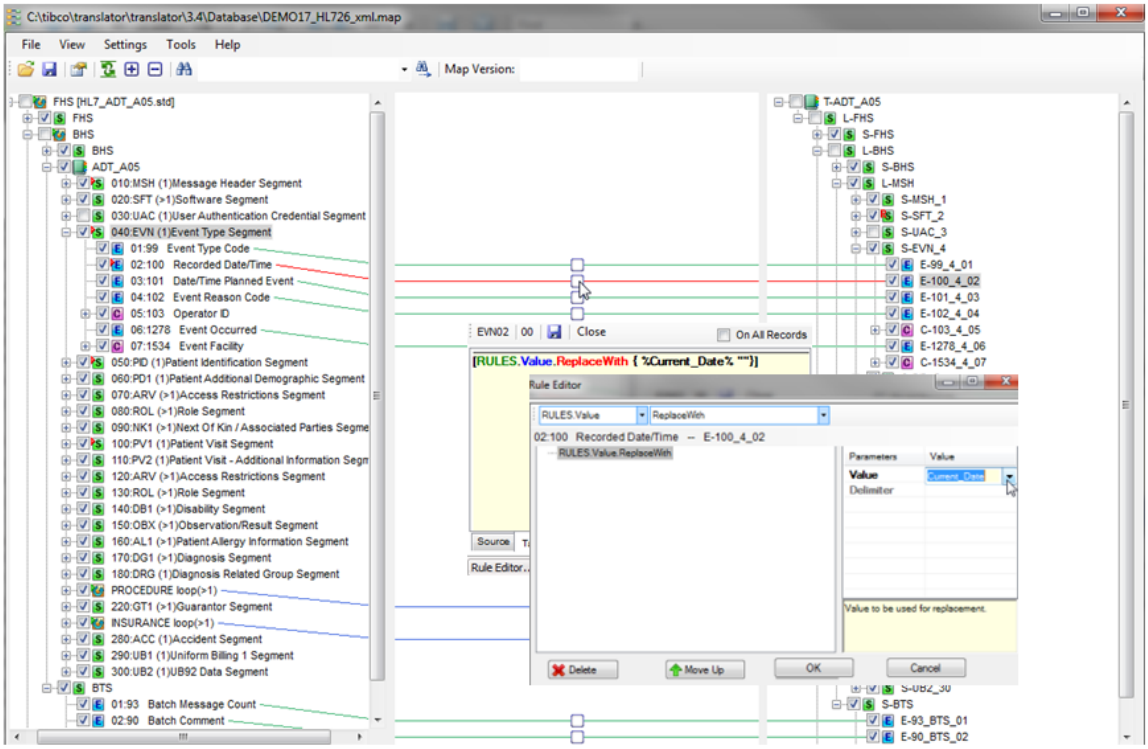
1. Open Translation Tool. You can do so in one of the following ways:

   - Select Start > All Programs > TIBCO_HOME > Translator in Windows.

- Click the  button in the TIBCO Foresight® EDISIM® Standards Editor toolbar.

2. Select **File > Open Source Guideline** in the menu. The Open Source Guideline dialog appears. Navigate to the directory to import the source guideline. This is the guideline that describes the input data.
3. Select **File > Open Target Guideline** in the menu. The Open Target Guideline dialog appears. Navigate to the directory to import the target guideline. This is the guideline that describes the output data.
4. Expand the source and target nodes to edit the map. Check the checkbox in front of the source data item from the left panel and then check the corresponding checkbox of the target data item from the right panel. If the mapping succeeds, there is a red or green line connecting the source and the target data.

   To delete the mapping between the source data and the target data, uncheck the checkboxes in front of the source data and the corresponding target data.
5. Set up transformation rules. Double-click the box as shown. The Rule Editor dialog appears. Edit rules based on your requirements in the dialog.



6. Test the mapping file to verify your results. Click the  button in the Translation Tool toolbar and then click the **Test** tab at the bottom of the Translation Tool window. In the Test tab, test the mapping.
7. Select **File > Save** in the menu to save the map in the Translator Database Directory.

HL7 Reference Information

The following information is provided as a reference for understanding HL7 messaging and how TIBCO products can be used to build solutions that meet a company's internal and external HL7 and HIPAA healthcare messaging needs:

Healthcare Messaging Standards

HL7 and HIPAA Healthcare Messaging Standards

HL7 (Health Level Seven) is a messaging standard for exchanging clinical and administrative data between healthcare applications from various vendors, typically within an enterprise. HIPAA (Health Insurance Portability and Accountability Act) was enacted in 1996 and is designed to streamline healthcare transactions across enterprises and to uphold patients' privacy rights.

TIBCO ActiveMatrix™ BusinessWorks Plug-in for HL7 can address the HL7 integration needs of an enterprise. In addition to addressing HL7 needs, it also addresses the HL7 requirement for integration with HIPAA, as HIPAA has HL7 requirements for some of its transactions. See TIBCO HL7 and HIPAA Solutions for more information.

How HL7 and HIPAA Messaging Can Overlap

Within enterprises like hospitals, events like patient admission and billing trigger HL7 messages between enterprise applications. Between enterprises like hospitals and insurance companies, which exchange HIPAA (Health Insurance Portability and Accountability Act) EDI-X12 transactions, HL7 is used in certain HIPAA EDI-X12 transactions.

For example, when a patient is admitted to hospital, the Patient Administration System (PAS) records details about the patient and the admission. Other systems, such as Pathology Laboratory Information Systems or Pharmacy Systems, are likely to need information about the new patient. The PAS sends an HL7 message about the patient to each of the appropriate hospital systems.

Later, when the hospital sends a claim for payment to another enterprise, like an insurance company, the hospital and the insurance company exchange HIPAA EDI-X12 messages. One or more of those HIPAA EDI-X12 messages may contain an embedded HL7 message with additional patient information.

HL7

HL7 is a structured, message-oriented framework for communications between healthcare application systems. When designers began to develop HL7, they borrowed from what is now ASC X.12 (EDI). However, the current similarity between the two standards is accidental for practical purposes since HL7 has evolved down its own path since its creation.

The HL7 protocol architecture is hierarchical, moving from high-level groupings and structures to a set of several hundred data fields. Each level of the hierarchy serves a different organizing purpose.

Functional Group

Areas of the protocol are grouped according to common application function. For example, ADT, Order Entry, Finance, Control, and Ancillary Reporting all represent groups described in the standard. Different functional groups are typically given individual chapters in the HL7 specification document.

Message Type

Within a functional group, one or more message types are defined that can be implemented in various combinations to support high-level business rules for the applications involved. For example, ADT only specifies one message type, while Order Entry describes more than a dozen.

Message Definition

Within each message type, one or more message definitions describe the specific set or combination of segments that make up a properly formed message. For example, ADT distinguishes among more than 30 separate message definitions based on trigger events or more detailed business rules. Each message definition includes one or more segments.

Segment Definition

Segments provide a logical grouping for data elements. For example, the Patient Identification (PID) segment includes fields for such identifying information as patient name, social security number, medical record number, account number, and miscellaneous demographic details. How fields are grouped in segments forms part of the HL7 implied data model. Segments can be required or optional, can be nested, and can repeat. A parsed message, then, can take on a relatively arbitrary yet unambiguous form. This is an important characteristic in the context of decoding and encoding messages.

Field

The HL7 standard identifies several hundred data elements for communicating patient demographic, clinical, and financial information. HL7 uses more than a dozen abstract data types to define the nature of the fields. One consequence is that some fields may hold more than one data element.

For example, a field that holds a time stamp (TS) follows a prescribed format. In addition, many fields are or can be coded, and the standard includes a variety of code tables to define acceptable contents. While each field is defined with a maximum length, the standard does not intend to prescribe format to that level of detail. In merely includes lengths because it helps readers understand the purpose of the field, and it may have a practical importance in specific implementations.

HIPAA and HL7 Messaging

HL7 messages are typically exchanged between enterprise applications like a Patient Administration System and a Pathology Laboratory Information system. However, HL7 messages must sometimes be embedded in certain HIPAA EDI-X12 transactions between enterprises like hospitals and insurance companies.

For example, when a hospital sends a claim for payment to another enterprise, like an insurance company, the hospital and the insurance company exchange HIPAA EDI-X12 messages. HIPAA EDI-X12 messages may contain an embedded HL7 message with additional patient information requested from the insurance company.

TIBCO HL7/HIPAA Solution

To enable the exchange of HIPAA messages with HL7 attachments between enterprises, TIBCO offers:

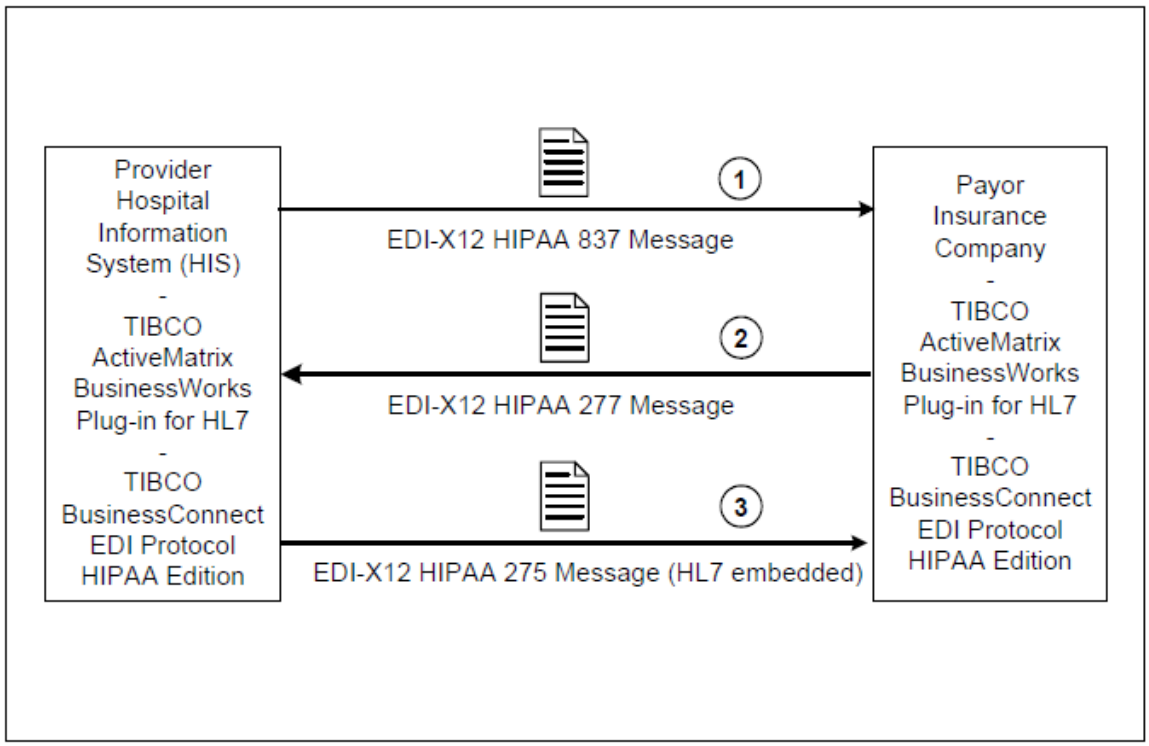
- TIBCO ActiveMatrix BusinessWorks™ Plug-in for HL7, the TIBCO implementation of the HL7 standard. It provides a framework for a healthcare provider's internal HL7 HIPAA messaging.
- BusinessConnect™ EDI Protocol HIPAA Edition powered by Instream®, the TIBCO implementation of the EDI standard. It offers specialized functionality for customers who require HIPAA compliance.

Together, these products enable the exchange of HIPAA messages and embedded HL7 messages between enterprises like hospitals and insurance companies.

Sample High-Level Scenario

These two products can be used in concert to provide the ability to include HL7 information in EDI-X12 275 HIPAA messages exchanged between a healthcare enterprise and an external payor enterprise.

The following figure shows how the message exchange occurs.



The following steps may occur:

1. You use BusinessConnect™ EDI Protocol HIPAA Edition powered by Instream® to send an EDI-X12 837 Health Care Claim HIPAA message to an external payor.
2. The payor uses BusinessConnect™ EDI Protocol HIPAA Edition powered by Instream® to send an EDI-X12 277 Health Care Information Status Notification HIPAA message back to you. This is a responder request for more information.
3. You use BusinessConnect™ EDI Protocol HIPAA Edition powered by Instream® to send an EDI-X12 275 Patient Information HIPAA message back to the payor. This is a responder response message.

   The `BIN` segment of this 275 XML message contains an embedded HL7 ORU (Observation Result/ Unsolicited) message created in TIBCO ActiveMatrix BusinessWorks™s Plug-in for HL7 with the additional patient information that was requested. This information can include diagnoses, test results, observations, and treatments.

   The `BIN` segment is carried in the response field of the private `RESPONDER.RESPONSE` message in EDI HIPAA. See the *TIBCO BusinessConnect™ EDI Protocol HIPAA Edition Guide* for more information on the `RESPONDER.RESPONSE` messaging class.The embedded HL7 ORU message is obtained during EDI processing in TIBCO ActiveMatrix BusinessWorks™. Also see See HL7 ORU Messages.

There are many other scenarios where it makes sense to use TIBCO ActiveMatrix BusinessWorks™ Plug-in for HL7 and BusinessConnect™ EDI Protocol HIPAA Edition powered by Instream® together. For example, after a patient is discharged, the HL7 discharge message can trigger queries to a billing system, which can then generate a proper 837 HIPAA message to send to the insurance company. This streamlines the operation for claim submission, brings valuable turn-around time for the provider, and helps increase efficiency and return on investment.

Provider and Payor Roles When Sending an EDI-X12 275 HIPAA Message

To submit an EDI-X12 275 HIPAA message, a **provider system** must do the following:

- Express patient data as an HL7 ORU message. See ref for information on ORU messages.
- Use the appropriate LOINC (Logical Observation Identifier Names and Codes) modifier code.
- Wrap the HL7 ORU message in the BIN segment of an X12N-compliant 275 claims attachment transaction. X12N is for exchanging insurance, eligibility, and managed care information.

To process an EDI-X12 275 HIPAA message, a **payor system** must be able to parse the HL7 ORU message embedded in the 275. TIBCO ActiveMatrix BusinessWorks™ Plug-in for HL7 can easily perform this task.

HL7 Messages

This section describes various aspects of HL7 messages.

HL7 Message Components

Each HL7 message type definition contains a set of lines known as segments. Each segment groups related information, and has a three-character name and a pre-defined format of specific fields. The fields are separated by the | (pipe) character, and may be further divided into subcomponents with the ^ character.

Different HL7 events trigger different message types - each message type has a defined set of segments that are joined together to provide all the required information regarding that event. Some segments are mandatory, and must be included in the message, and other segments are optional.

Each segment is delimited by one of the following end line characters:

- \n
- 0x0D

Each segment contains a specific type of information. Examples:

- MSH. Information about the sender and receiver of the message, type of message, and time stamp.
- PID. Demographic information. This includes patient name, id codes, and address
- PV1. Information on the patient's hospital stay. This includes location, and referring doctor.

Each segment in turn contains a sequence of fields that are composites. These fields may or may not repeat. Composites are delimited by "|".

HL7 Message Definition and Sample

In TIBCO Foresight® Instream® and TIBCO Foresight® Translator, each message definition contains the following parts:

- Segment Grammar. This gives the HL7 format of the message.
- Table Grammar. This gives the internal view of the message as seen by an enterprise application.
- Identity. This determines how TIBCO Foresight® Instream® and TIBCO Foresight® Translator recognizes the message.

The following is a typical HL7 ADT^A04 message, which is sent after a patient enters the hospital. When the patient information is entered into the Patient Administration System (PAS), the information is sent to other enterprise systems, such as Pathology Laboratory Information Systems or Pharmacy Systems.

```
MSH|^~\&|EPIC|EPICADT|SMS|SMSADT|199912271408|CHARRIS|ADT^A04|1817457|D|2.3|
EVN|A04|199912271408|||CHARRIS
```

```
PID||0493575^^^2^ID 1|454721||DOE^JOHN^^^^|DOE^JOHN^^^^|
19480203|M||B|254 E238ST^^EUCLID^OH^44123^USA||(216)731-4359|||
M|NON|400003403~1129086|999-|

NK1||CONROY^MARI^^^^|SPO||(216)731-4359||EC|||||||||||||||||||||||||

PV1||O|168 ~219~C~PMA^^^^^^^^^|||277^ALLEN FADZL^BONNIE
^^^^|||||||||| ||2688684|||||||||||||||||||||||
199912271408||||||002376853
```

Making HL7 Messages More Readable

HL7 messages are in ASCII, and are difficult to interpret. However, you can use TIBCO Foresight® Instream® and TIBCO Foresight® Translator to convert an HL7 ASCII message into a tree format. You can also use TIBCO Foresight® Instream® and TIBCO Foresight® Translator to convert an HL7 ASCII message to XML. TIBCO has defined an XML format that is easy to read. This also allows data to work well with other components in TIBCO ActiveMatrix BusinessWorks™.

You can also use TIBCO Foresight® Instream® and TIBCO Foresight® Translator to convert the XML message back to ASCII.

A HIPAA-compliant ORU message contains the following:

- Message header and patient information
- One or more observation request (OBR) segments defining observation type and request specifics.
- One or more observation result (OBX) segments defining, quantifying, and qualifying the results.
- One or more LOINC (Logical Observation Identifier Names and Codes) modifier codes used in the OBR and/or OBX segments as universal identifiers for laboratory and other clinical observations.

TIBCO HL7 and HIPAA Solutions

Solutions

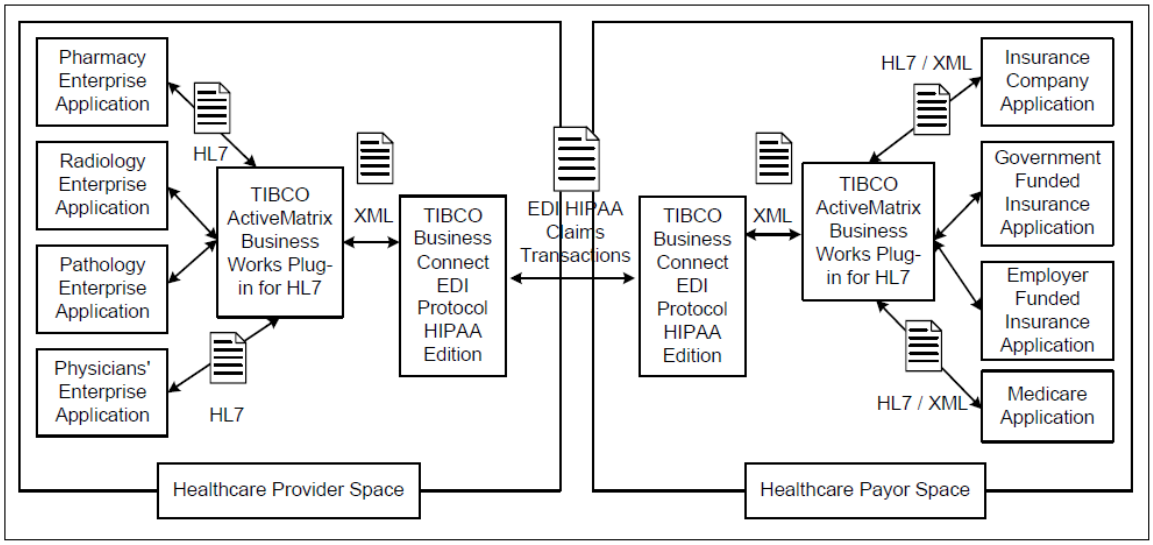
The following TIBCO products work together to provide a solution to a company's internal and external HL7 and HIPAA healthcare messaging needs:

- TIBCO ActiveMatrix BusinessWorks™ Plug-in for HL7 - the implementation of the HL7 standard with TIBCO ActiveMatrix BusinessWorks™. It provides a framework for quick integration of a healthcare provider's HL7 systems and offers HL7 message customizations together with business process design and management.

  TIBCO ActiveMatrix BusinessWorks™ for HL7, can receive inbound and outbound HL7 messages and convert data back and forth between HL7 and XML. The Interface Engine can then make business decisions by looking at the data in the messages. The Interface Engine also provides centralized message routing.
- BusinessConnect™ EDI Protocol HIPAA Edition powered by Instream® - the TIBCO implementation of the EDI standard, and offers specialized functionality for customers who require HIPAA compliance.

  BusinessConnect™ EDI Protocol HIPAA Edition powered by Instream® enables the exchange of HIPAA messages and embedded HL7 messages between enterprises like hospitals and insurance companies.

  The following shows an overview of how HIPAA and HL7 transactions are exchanged in the context of the products described above.



- TIBCO Foresight Products

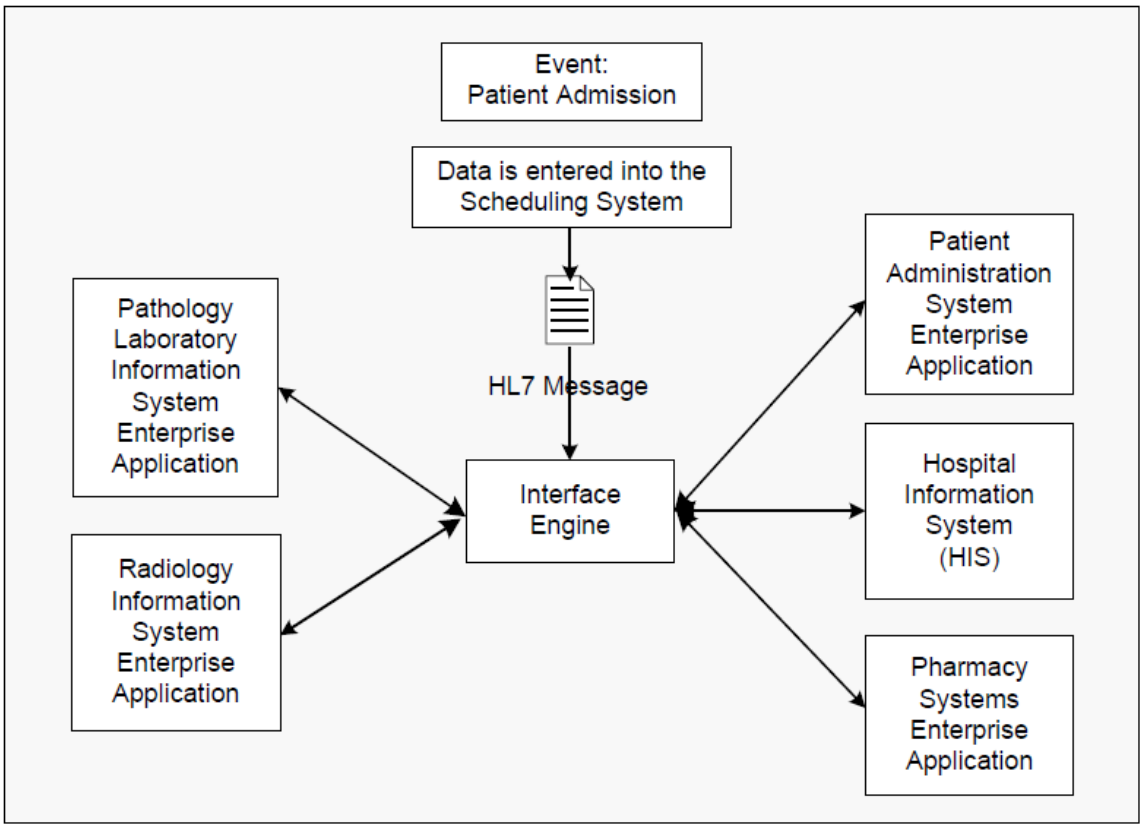
  TIBCO Foresight® Translator and TIBCO Foresight® Instream® act as the HL7 parser engine. TIBCO Foresight® Translator is a specialized, high-speed transformation engine that enables the mass conversion of transaction files based on pre-built or custom maps, providing direct translation to and from HL7, XML, and Flat File formats with no interim staging required.

  TIBCO Foresight® Instream® ensures the compliance of inbound and outbound data using the fastest and most thorough transaction validation engine.

  TIBCO Foresight® EDISIM® is a design-time tool that allows you to create and edit HL7 message definitions based on your needs, validate HL7 messages based on the message definition and create STD, MAP, XML, and XSD files

Architecture

This figure shows an example of how the key components, such as an interface engine implemented with TIBCO ActiveMatrix BusinessWorks™ Plug-in for HL7 on top of TIBCO ActiveMatrix BusinessWorks™, fits into a hospital's messaging architecture. It illustrates how an event like a patient admission would trigger message flow to the relevant systems and components.


Event:
Patient Admission
Data is entered into the Scheduling System
HL7 Message
Pathology Laboratory Information System Enterprise Application
Radiology Information System Enterprise Application
Interface Engine
Patient Administration System Enterprise Application
Hospital Information System (HIS)
Pharmacy Systems Enterprise Application