TIBCO BusinessEvents® Plug-in for EDI

Developer's Guide

Software Release 2.0.0
October 2012

Two-second advantage®

Important Information

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

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

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

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

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

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

Please see **TIBCO <*product*> <*edition*> <*version*> TIBCO EULA and TPS Notices.pdf** for licensing information.

Contact Information

TIBCO Software Inc., Foresight Group
655 Metro Place South
Suite 900
Dublin OH 43017
Phone: (614) 791-1600
Fax: (614) 791-1609
Web: http://foresight.TIBCO.com
Support E-mail: FSsupport@tibco.com

Contents

Overview

About the TIBCO BusinessEvents® Plug-in for EDI

BusinessEvents® is a decision management platform that provides intelligent routing of data and end-to-end state management. The BusinessEvents® Plug-in for EDI allows users to access Foresight technology for EDI through the BusinessEvents Platform:

- TIBCO Foresight™ Instream® – validates transactions according to industry standards, organizational guidelines, and specific business rules.
- TIBCO Foresight™ Translator – provides source-to-target mapping to and from EDI, XML, and Flat File.

At this time the Business Events (BE) Plug-in for EDI supports 270 and 271 transaction processing.

This document details the use of Instream® and Foresight™ Translator within the BE framework.

Functionality

The Business Events (BE) Instream Plug-In models the transaction processing between a clearing house and adjudication system for 270 and 271 EDI transmissions, two of the so-called the High Volume EDI transmission types. The plug-in's primary functions are:

1. Route messages containing the 270 and 271 EDI between the clearing house and the adjudication system;
2. Validate the 270 and 271 EDI passed in from the clearing house and the adjudication system.
3. Compose and send metric messages to a queue that can be used as a data source for a viewer/analysis engine.

The product contains sufficient functionality to implement the 270/271 solution, and it tries to minimize assumptions about how a user might use it in a production environment. Customers can customize the BE project to implement 270/271 solutions that conform to their environments.

Audience

This document assumes that users of this product will be developers with the following skills:

- **Business Events.** Must be proficient in BE IDE.
- **EMS.** Need to start and manage queues.
- **EDI.** Must understand basic idea behind the 270/271 transmissions, clearing house, adjudication system, validation, translation.
- **OS environment.** Must be able to configure the OS environment to satisfy run-time dependencies for Java.

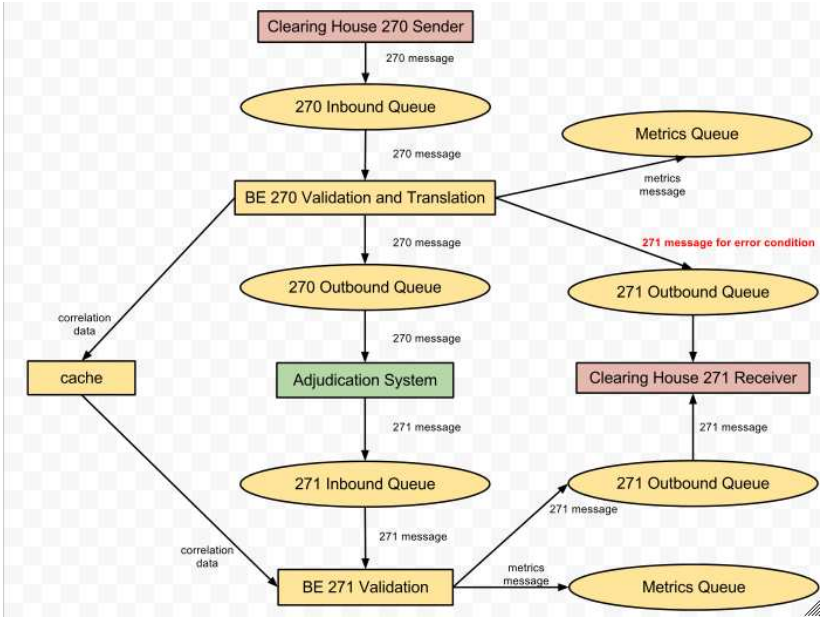
Data Flow

Overview of Data Flow

The clearing house sends a 270 EDI transmission to BE. Using Instream, BE performs a validation of the EDI and on success it sends the 270 to the adjudication system. The adjudication system performs its work and sends a 271 to BE which validates it using Instream and then passes the 271 to the clearing house.

The application can be configured to write a metrics message to an EMS queue. This message will contain data about the round trip processing of the 270 and 271 EDI.

All data are represented as messages on JMS-compliant queues in EMS. That means the 270 from the clearing house will go onto a queue, as well as the 271 that gets sent back to it. The same is true for the adjudication system; the 270 will go onto an EMS queue and the adjudication system will put the 271 onto an EMS queue. All of these queues are accessible to BE.



Error Handling

An error can occur if

- the data in the inbound 270 or 271 message is invalid
- the payload fails validation or translation

  - validation results can be tested against configurable thresholds
- there is a correlation error.

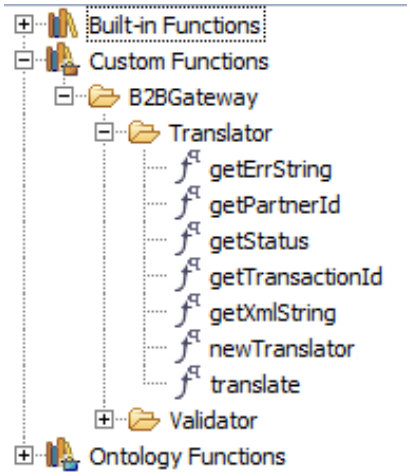
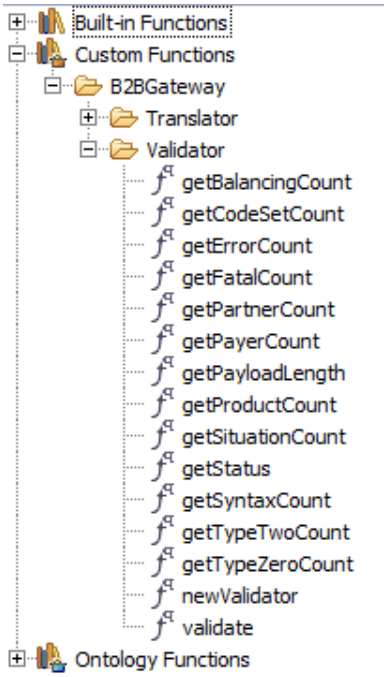
BE will put a message on the outbound queue for 271s when an error occurs. This message includes the properties statusCode and errorResponseMessage. The payload will contain the EDI that caused the error condition. The metrics will also reflect that the round trip failed.

Components

InstreamPlugIn.jar

This file contains the basic functionality for the Instream validation and translation engines that BE accesses via the custom functions.

The jar file contains the file *functions.catalog*, which defines the functions that will be exposed for Instream and Translator. The function classes under Custom Functions will align with the definitions in the *functions.catalog* file (figure does not necessarily show the final methods)

Translator Functions	Validator Functions
	

Validator Functions

Function	Description
newValidator	Creates an instance of a Instream Validator object
validate	Validates an EDI transmission
getStatus	Gets the status of the validation procedure. A status of 100 indicates success.
get<errorOrSeverity>	Returns a count of the following severity levels and errors from Instream: Balancing, CodeSet, Partner, Payer, Product, Situation, Syntax, TypeTwo, TypeZero, Error, and Fatal.

Translator Functions

Function	Description
newTranslator	Creates an instance of a Translator object
translate	Translate an EDI transmission
getStatus	Gets the status of the Translator procedure. A status of 100 indicates Success.
getTransactionID	Extracts the transaction key for the 270 and 271 EDI documents.
getPartnerId	Extracts the partner key.

BE Project EDI_27x_Instream

Overview

The BE Project EDI_27x_Instream project contains the base functionality for performing validation, translation, and routing the messages between the clearing house and adjudication system.

Naming conventions for Ontology

To keep things somewhat easy to identify, the various related objects in the ontology use the same root name wherever possible and are distinguished from one another by a suffix denoting the ontology type. For example, the destination that handles the inbound 270 events is named *inbound_270_D*. Its default event type is named *inbound_270_E*. The preprocessor rule function for this event type is named *inbound_270_PP* and the rule that handles this event type is named *inbound_270_R*.

Channels, Destinations, Queues, and Events

There is one channel named *JMS*. The destinations, events, and queues are named for the direction of the message and the EDI transmission type. Direction is relative to BE. All of the events inherit from *EDI_Base.event*.

Destination	Event	EMS Queue	Function
inbound_270_D	inbound_270_E	BEInstream_270_Inbound	Handles the inbound 270 from the clearing house
outbound_270_D	outbound_270_E	BEInstream_270_Outbound	Handles the outbound 270 to the adjudication system
inbound_271_D	Inbound_271_E	BEInstream_271_Inbound	Handles the inbound 271 from the adjudication system
outbound_271_D	outbound_271_E	BEInstream_271_Outbound	Handles the outbound 271 to the clearing house
outbound_metrics_D	outbound_scorecard_E	BEInstream_Metrics	Handles metrics messages

Queues are described in further detail below.

Rules and Rule Functions

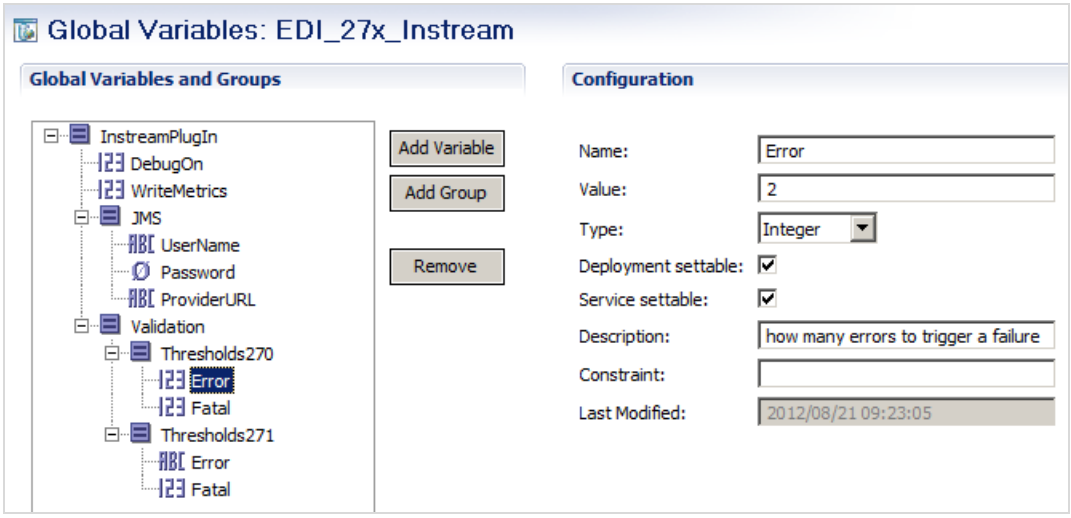
There are 2 rules. *inbound_270_R* takes an *inbound_270_E* event and creates the *Transaction* concept that tracks the end to end processing. *EDI_Functions_R* handles the translation and validation of the EDI, and routing of the events between the adjudication system and the clearing house.

There are several groups of rule functions.

Package Name	Description
AppLogging	Contains functions for creating and writing log messages.
PreProcessor	The pre-processor rule functions validate the data in the message for completeness and assert the events if the data is valid.
Metrics	Functions for handling the metrics data.
ValidateAndTranslate	Contains several functions used for validating and translating the EDI.

Global Variables

The project uses the following global variables.



Global Variable	Description
DebugOn	Set this to 1 to enable debugging messages. Set this to 0 to suppress them.
WriteScorecard	Set this to 1 to enable writing metrics messages to BEInstream_Metrics. Set this to 0 to suppress them.
JMS/UserName	User name for EMS account
JMS/Password	Password for EMS account
JMS/ ProviderURL	URL for EMS service
Validation/Thresholds270 Validation/Thresholds271	This is a set of threshold values that allows the user to indicate how many validation errors of specific types he is willing to tolerate before he declares an EDI document as invalid.

Threshold Values

These values are used to determine if processing should continue when there are type 3 (Error) or type 4 (Fatal) error conditions. If the value returned by the validation step is greater than the threshold value, then an error condition will occur and processing will stop. A message will be placed on the outbound 271 queue with properties describing the error. The *statusCode* property will be a value that is not 100. The *errorResponseMessage* will describe the problem.

Messages and Queues in EMS

All Messages that are placed on queues are instances of the class *TibjmsTextMessage*. Except for the metrics message, each message will have a *payload*, which will contain either the 270 or 271 EDI.

In naming the queues, the notion of *inbound* and *outbound* is always from the BE point of view.

All messages will have a *requestExtId* filed for correlation. This is a unique value sent by the clearing house that identifies the request.

Queue	Named Fields	Payload	Description
BEInstream_270_Inbound	requestExtId	270 EDI	This queue handles the inbound 270's from the clearing house.
BEInstream_271_Outbound	requestExtId transactionKey errorMessage statusCode	Validated 271 EDI	The queue handles the outbound 271's from BE to the clearing house. If *statusCode* returned in this message is 100, then the operation was successful. Otherwise there will be text in the errorMessage.
BEInstream_270_Outbound	requestExtId transactionKey	Validated 270 EDI	This queue handles the outbound 270's from BE to the adjudication system.
BEInstream_271_Inbound	requestExtId transactionKey	271 EDI	This queue handles the inbound 271's from the adjudication system to BE.
BEInstream_Metrics	requestExtId transactionKey <metrics, discussed below>	None	This queue holds metrics data. The user is free to manage this.

When a 270 is translated after validation, the rule will pull out the *transaction key* from the EDI and pass it to the adjudication system as part of the outbound 270 message. This is represented as the string <*transaction id*>-<*last name*>-<*subscriber id*>.

270 Transaction Key SubItem	EDI Path
<transactionId>	270/BHT03
<lastName>	270/2000A/2000B/2000C/2100C/NM103
<subscriberId>	270/2000A/2000B/2000C/2100C/NM109

Metrics

BE tracks the messages as they move through validation, translation, and adjudication. For each 270 request with a non-null requestExtId and non-null payload, a message will be written to the EMS queue *BEInstream_Metrics*. The message will contain the following types of data:

- Identification, such as the requestExtId used for correlation
- Start times of processing and adjudication
- Durations for end-to-end processing, adjudication phase, validation, translation steps.
- Payload sizes for EDI
- Outcomes for validation, translation, and round trip
- Counts for various Instream error types and severities.

Where it is necessary, separate metric values exist for both the 270 and 271 parts of the processing.

Please see the Appendix for an example of how to create a database table that could handle the metrics in the message.

Installation and Setup

The product consists of a ZIP File that contains the InstreamPlugIn.jar file and the BE Project *EDI_27x_Instream*.

System Software

The OS must be 64-bit.

- TIBCO Foresight Instream 64 bit
- TIBCO Foresight Translator 64 bit
- TIBCO EMS 64 bit
- TIBCO Business Events 5.0 64 bit
- Java 1.7

Jar Files

InstreamPlugIn.jar contains the custom functions that BE will use to validate and translate EDI. It has dependencies on fstransapi.jar and instreamapi.jar.

Jar File	Source	Destination
InstreamPlugIn.jar	Ships with product	<tibcoRoot>/be/ <version>/lib/ext/tpcl
fstransapi.jar	Ships with Translator in the Java folder	<tibcoRoot>/be/ <version>/lib/ext/tpcl/lib
instreamapi.jar	Ships with Instream in the Java folder	<tibcoRoot>/be/ <version>/lib/ext/tpcl/lib

If the *lib* folder does not exist beneath the *tpcl* folder, create it. Note that Translator can be installed in the Instream installation directory structure.

Logging

Application level log messages can be captured from 2 sources: the BE rules/rule functions and the Custom Functions.

Logging for the Custom Functions can be enabled by setting the environment variable *InstreamPlugInFileName* to point to the log message file. Also, you can set this as a system property.

Logging in the rules and rule functions can be enabled by setting the global variable *DebugOn* to the value *1*. This will cause messages to be written using the *System.debugOut* function. To suppress log messages in the rules and rule functions, set this variable to 0.

EMS Setup

You will need to create the following queues in EMS: BEInstream_270_Inbound, BEInstream_271_Outbound, BEInstream_270_Outbound, BEInstream_271_Inbound, BEInstream_Metrics.

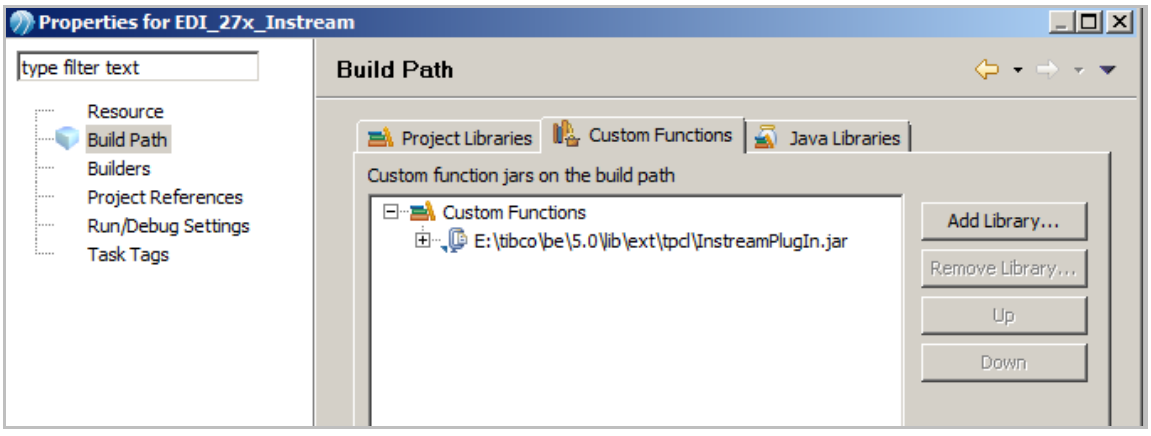
BE Setup

JMS

Set up BE so that it has the jar files necessary to handle the TIBCO JMS messages residing on EMS queues. Please refer to the BE documentation for a description of what needs to be done here.

Custom Functions

You will need to adjust the location of the InstreamPlugIn.jar file so that it points to where you have installed it.



be-engine.tra

For the property *tibco.env.PATH* in be-engine.tra, add the names of the *bin* folders for Instream and Translator.