



ibi™ iWay® Service Manager

iWay Integration Solution for TRADACOMS User Guide

Version 9.2.0 | November 2024

Contents

Contents	2
Introducing the iWay Integration Solution for TRADACOMS	6
Prerequisites for TRADACOMS	6
A Brief History of Electronic Data Interchange	6
Early Standardization Efforts	7
The TRADACOMS Standard	7
Components of a TRADACOMS Transmission and File Structure	8
Features of the iWay Integration Solution for TRADACOMS	9
TRADACOMS Syntax and Transmission Structure	10
Components of the iWay Integration Solution for TRADACOMS	11
Ebix	11
Listener	12
Preparser	12
Validation Report	12
Installing the iWay Integration Solution for TRADACOMS	12
Install the iWay Integration Solution for TRADACOMS	13
Deployment Information for Your iWay Integration Solution	15
iWay Service Manager	15
iWay Correlation Facility	15
Using a Channel to Construct a Message Flow	16
Components of a Channel	17
Working With TRADACOMS Inbound and Outbound Applications Using iWay Integration Tools	18
TRADACOMS Inbound and Outbound Applications Overview	18
TRADACOMS Inbound and Outbound Applications Prerequisites	18

Extracting TRADACOMS User Samples	19
Extract User Samples for TRADACOMS	19
Importing TRADACOMS User Samples to iIT as a Workspace	21
Import TRADACOMS User Samples to iIT as a Workspace	21
Publishing iIAs to the iSM Registry	27
Publish iIAs to the iSM Registry	27
Deploying iWay Integration Applications to iWay Service Manager	30
Deploy iWay Integration Applications to iWay Service Manager	30
Configuring Special Registers	33
Configure Special Registers	33
Stopping Inbound and Outbound Processing	39
Stop Inbound (TRADACOMS to XML) Processing	39
Stop Outbound (XML to TRADACOMS) Processing	39
Testing Sample Applications	40
Test the Sample Inbound (TRADACOMS to XML) Application	40
Test the Sample Outbound (XML to TRADACOMS) Application	45
Inbound Processing: TRADACOMS to XML	51
Inbound Processing Overview	51
Special Register Sets	52
Modify SREG Sets in the Channel	53
Sample Channel File Listener	54
Sample Channel Preparser	55
Sample Channel Process Flow	56
Adding an Ebix to a Channel	56
Add an Ebix to a Channel	56
Rebuilding Your Application	57
Rebuild Your Application	57
Outbound Processing: XML to TRADACOMS	60
Outbound Processing Overview	60

Special Register Sets	61
Modify SREG Sets in the Channel	62
Sample Channel File Listener	62
Sample Channel Process Flow	64
Adding an Ebix to a Channel	65
Add an Ebix to a Channel	65
Rebuilding Your Application	65
Rebuild Your Application	66
Ebix-Supported Transaction Set	69
TRAD93	69
Configuring the EDI Activity Driver	70
EDI Activity Driver Overview for TRADACOMS	70
Configuring the EDI Data Provider	70
Configure the EDI Data Provider	70
Configuring the EDI Activity Driver	73
Configure the EDI Activity Driver	73
Using iWay Integration Tools to Configure an Ebix for TRADACOMS	80
Ebix Overview	80
Ebix Prerequisites	80
Working With iWay Integration Tools (iIT)	81
Import an Ebix	81
Edit an Ebix	86
Export an Ebix	89
Sample TRADACOMS Files	93
TRADACOMS ORDHDR (Order Header)	93
TRADACOMS INVFIL (Invoice File Header)	93
ibi Documentation and Support Services	95

Legal and Third-Party Notices **96**

Introducing the iWay Integration Solution for TRADACOMS

The iWay Integration Solution for TRADACOMS transforms TRADACOMS documents into standard XML format, or transforms XML representations into TRADACOMS format.

This section provides an overview of TRADACOMS and describes the features that are provided by the iWay Integration Solution for TRADACOMS.

Prerequisites for TRADACOMS

Before you use the iWay Integration Solution for TRADACOMS for inbound (TRADACOMS to XML) and outbound (XML to TRADACOMS) processing, ensure that the following prerequisites are met:

- You have a working knowledge of iWay Service Manager (iSM).
- You have a working knowledge of iWay Integration Tools (iIT).
- iSM Version 8.0 is installed.
- iIT Version 8.0 is installed.
- iWay Integration Solution for TRADACOMS (Patch) is installed.
- System and channel Special Registers (SREGs) are updated to match your directory structure.

A Brief History of Electronic Data Interchange

Electronic Data Interchange (EDI) is a set of standards for formatting information that is electronically exchanged between one business and another, or within a business. These standards describe how documents for conducting certain aspects of business—such as purchase orders and purchase order acknowledgments—are structured.

By specifying a standardized, computer-readable format for transferring data, EDI enables the automation of commercial transactions around the world. It provides a common, uniform language through which computers can communicate for fast and efficient transaction processing.

Early Standardization Efforts

Before the development of standards, many businesses used proprietary systems to exchange trading information such as purchase orders and invoices. However, they recognized the economic need for a faster, less costly way to process information in order to stay competitive in the business world. Business sectors such as transportation, grocery supply, and banking drove the creation of standards for the communication of data.

In 1968, the United States Transportation Data Coordinating Committee (TDCC) was formed to oversee the design and development of format standards for transportation documents. In 1975, the TDCC released its first standard, the Rail Transportation Industry Application.

The Rail Transportation Industry Application focused on the content of a message—rather than the means of transmission—through the use of transaction sets. A *transaction set* is a business document that consists of an arrangement of data segments. The data segments include data elements in an exact order. The concept of the transaction set is the basis of the EDI ANSI X12 standard created later and widely used today.

About the same time that the TDCC was formed, the United Kingdom (UK) started its own effort to develop standard transaction documents for trans-Atlantic trade. The UK Department of Customs and Excise, with the help of the British Simplification of Trade Procedures Board (SITPRO), developed a competitive document standard for international trade, named TRADACOMS.

The TRADACOMS Standard

TRADACOMS is an early standard for EDI used in the UK retail and grocery sector. It was introduced in 1982 as an implementation of the UN/GTDI syntax, one of the precursors of EDIFACT, and was maintained and extended by the UK Article Numbering Association (now called GS1 UK). The standard is obsolescent since development of it effectively ceased in 1995 in favor of the EDIFACT EANCOM subsets. Despite the fact that further development of TRADACOMS effectively ceased in 1995, it has proved durable and the majority of the retail EDI traffic in the UK still uses TRADACOMS.

Components of a TRADACOMS Transmission and File Structure

A TRADACOMS transmission consists of:

- A Start of Transmission Segment (STX)
- One or more messages
- An optional Reconciliation Message (RSGRSG)
- An End of Transmission Segment (END)

A TRADACOMS File consists of a series of Messages. Each Message has a similar overall structure, beginning with a Message Header Segment (MHD) and ending with a Message Trailer Segment (MTR). The contents, however, are different for each transaction type or File Format (for example, Orders, Invoices), and these are specified in the individual format specifications.

Normal practice should be to send one file per STX envelope. However, in library implementations, when several libraries trade through a shared EDI gateway, it is more efficient to send multiple files of the same type in a single STX envelope. This practice should be agreed between trading partners.

Each segment consists of:

- Segment label (for example, STX)
- Separator (=)
- One or more Data elements
 - Data (coded or free-text as permitted by rules)
 - Separator (:)
 - Data
 - Data Element terminator (+)
- Segment terminator (')

For example:

```
STX=ANAA:1+5012345678987:LIB+5098765432123:SUPP+01906+246359++BTOERS2'
```

The five characters that are listed and described in the following table have special significance in a TRADACOMS transmission.

Character	Description
=	The <i>equals sign</i> is used to separate a segment label from the first data element in the segment.
+	The <i>plus sign</i> is used to separate successive data elements within a segment.
:	The <i>colon</i> is used in composite data elements to separate successive component sub-elements.
'	The <i>apostrophe</i> is used to terminate a segment.
?	The <i>question mark</i> is used as a "release character". When any of these five characters with special meaning is required as part of the text content of a data element, it must be preceded by the release character to ensure that it is not misinterpreted. For example, <i>O'REILLY</i> must be sent as <i>O?'REILLY</i> . Note that the release character is not counted as part of the length of the data element.

Features of the iWay Integration Solution for TRADACOMS

The standards-based iWay Integration Solutions for Electronic Commerce reduces the amount of effort it takes to integrate document transformations with your internal

enterprise applications and third-party trading partners. It includes conversion and validation of documents to XML format, making it easy to include documents in your XML-based integration projects.

Features include:

- Integration with iWay Service Manager (iSM).
- Integration with iWay TPM (iTPM).
- Integration with more than 200 other information assets, including popular back-office systems, databases, and front-office systems.
- Integration with leading application servers. Some of the popular platforms are BEA WebLogic, IBM WebSphere, Sun Java Enterprise System, and Oracle Application Server.
- Support for synchronous and asynchronous bi-directional interactions for documents between application servers, integration brokers, third-party software packages, and messaging services.
- *Out of the box* Ebix-based support for transaction sets. For more information, see [Ebix-Supported Transaction Set](#).
- A Reusable framework for parsing, transforming, and validating documents without the need to write custom code.
- A data dictionary approach to facilitate transformations to XML. All iWay Integration Solutions use dictionaries to transform data from standard format to any other format, or from any format to standard format. They support flat files, comma-delimited files, popular relational database formats, XML, and more.
- Pre-built data dictionaries, XML schemas, transformation templates, and rule files for automatic transformation and validation of input and output documents.

TRADACOMS Syntax and Transmission Structure

The syntax is a looping data structure. Elements are variable length. Here are some other highlights:

- Batches begin with an STX segment and terminate with an END segment.

- The segment tag delimiter is an equal sign character (=) rather than a data element separator.
- Only implicit decimals are used.
- All dates use a six-digit format (YYMMDD).

Any segment can occur only once in a TRADACOMS message definition. Segments tend to be very specific with a qualifier to identify their function.

TRADACOMS uses *Files* with one or more examples of the message being preceded by a header message, and followed by one or more trailer messages. This avoids the duplication of common header and trailer information which can occur in other standards.

TRADACOMS files are only intended to be used within the UK. They make no allowance for currencies other than Sterling. Tax information is also geared to UK requirements.

For more information, see the sample documents in [Sample TRADACOMS Files](#).

Components of the iWay Integration Solution for TRADACOMS

iWay business components used in the construction of a message flow for TRADACOMS transactions include:

- e-Business Information Exchange (Ebix) File
- Listener
- Parser
- Validation Report

Ebix

iWay Software provides the various Ebix files to use in conjunction with iWay integration solutions.

For more information on the supported documents and transaction set, see [Ebix-Supported Transaction Set](#).

An Ebix is a collection of metadata that defines the structure of data.

Each document includes:

- A data dictionary, which describes the segments and elements that compose each document. The dictionary is used to validate and transform per the standard.
- An XML schema.
- A TRADACOMS to XML transformation template.
- A rule file, which validates business rules as defined by the standard or the user.

Listener

A listener is an iWay business component, which picks up an incoming message on a channel. The sample channel includes a File listener, which acquires transactions placed into a specified directory on a file system.

Preparser

A preparser is an iWay business component that converts incoming messages into XML documents.

Validation Report

A validation report service is created as an XML document in the data flow. The report consists of the input message, the output message, and a section that indicates *SUCCESS* or *ERROR*, and any error message from transformation. On *ERROR*, this report can be routed for inspection, remediation, and possible reprocessing.

Installing the iWay Integration Solution for TRADACOMS

For more information on the files and components described in this section (including installation instructions), see the *i706000SM1394 Patch Release Notes*.

The iWay Integration Solution for TRADACOMS includes the following files for installation:

- **iwtrad.jar**. A new .jar file.
- **iwtransc.jar**. A modified .jar file.
- **TRAD_TRAD93.ebx**. Ebix file for the supported TRADACOMS transaction set.
- **TRADACOMS_accelerator.zip**. This .zip file contains a predefined directory structure that is used by inbound and outbound channels to route documents as they are processed. All iWay E-Commerce adapters share a common paradigm for these directories.

Install the iWay Integration Solution for TRADACOMS

Procedure

1. Unzip the *TRADACOMS_accelerator.zip* file to a location on your file system. For example:

```
C:\TRADACOMS_accelerator
```

i **Note:** You must set the *TRAD_Installdir* Special Register (SREG) in the iSM Administration Console to reflect this location.

2. Stop iWay Service Manager (iSM).
3. Install patch *i706000SM1394*.

You can also manually copy the following .jar files to the `<ism_home>\lib` directory:

- iwtrad.jar
- iwtransc.jar

4. Start iSM.
5. Deploy and start the iWay Integration Application (iIA) containing the sample channels using iWay Integration Tools (iIT).

For more information, see [BAD XREF HERE](#) "Import the Archive for Sample Inbound

and Outbound Channels.

Deployment Information for Your iWay Integration Solution

This topic describes the iWay products used with your iWay Integration Solution for TRADACOMS and provides a roadmap to full information on those products.

It also introduces the concept of a channel for the construction of a message flow in iWay Service Manager.

iWay Service Manager

iWay Service Manager (iSM) is the heart of the Universal Integration Framework and is an open transport service bus. iSM uses graphical tools to create sophisticated integration services without writing custom integration code by:

- Using metadata from target applications
- Transforming and mapping interfaces
- Managing stateless processes

Its capability to manage complex integration interactions makes it ideally suited to be the foundation of a service-oriented architecture.

For more information, see the *ibi™ iWay® Service Manager User Guide*.

iWay Correlation Facility

The iWay Correlation Facility (also known as the Correlation Manager) maintains records of anticipated activities occurring in the system. Correlation actions take the correlation from OPEN to CLOSED state, and allow history to be recorded. Agents are provided to implement Correlation Facility interactions within process flows, however, it is possible to use this API to accomplish this same purpose within your own exits.

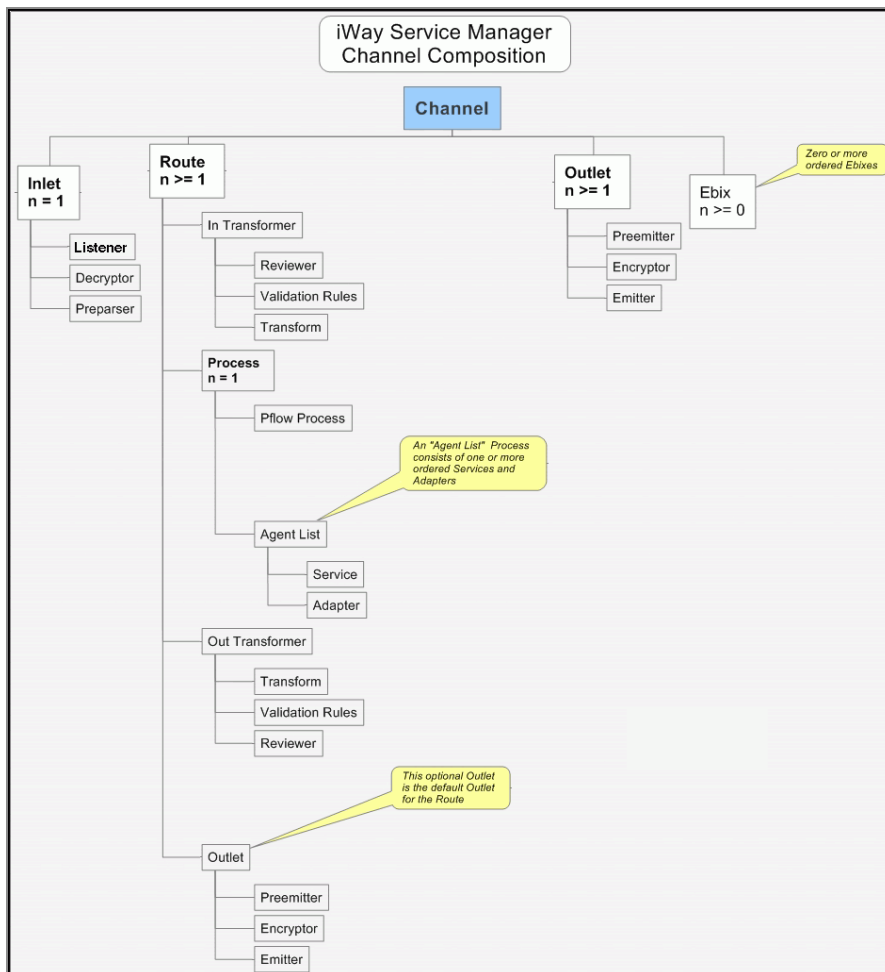
For more information on using the iWay Correlation Facility, see the *ibi™ iWay® Service Manager User Guide* and the *ibi™ iWay® Service Manager Programmer's Guide*.

Using a Channel to Construct a Message Flow

The following diagram shows the channel components available in the construction of a message flow.

The value *n* underneath a component name indicates how many instances of that component occur in a channel configuration (for example, zero, one, or more than one).

The required components are shown in **bold**.



Components of a Channel

A channel consists of:

- An inlet, which defines how a message enters a channel.
- A route, which defines the path a message takes through a channel.
- Outlets, which define how transformed messages leave a channel.
- Ebix files, which are collections of metadata that define the structure of data.

iSM provides a design-time repository called the Registry, where you assemble and manage the components in a channel.

Inlets contain:

- A listener, which brings the incoming message into the channel.
- Decryptors, which apply a decryption algorithm to an incoming message and verifies the security of the message.
- Parsers, which convert incoming messages to XML.

Routes contain:

- Transformers
- Reviewers
- Validation rules processors
- Process flows (pflow). This stateless, lightweight, short-lived microflow is executed to carry a message through processing. Pflows are created using iWay Integration Tools (iIT) and are persisted to the Registry.
- Java services, which handle the business logic of processing a message.
- Adapters, which are used to connect to back-end systems.

Outlets contain:

- Premitters.
- Encryptors.
- Emitters.

For more information on all of these components, see the *ibi™ iWay® Service Manager User Guide*.

Working With TRADACOMS Inbound and Outbound Applications Using iWay Integration Tools

This chapter describes how to work with TRADACOMS inbound and outbound applications using iWay Integration Tools (iIT).

TRADACOMS Inbound and Outbound Applications Overview

This chapter provides instructions to create, import, export, and work with TRADACOMS inbound and outbound applications.

The previously deployed iIA and channels will be used to transform TRADACOMS to XML for inbound processing and XML to TRADACOMS for outbound processing.

The inbound application channel creates an XML representation of a TRADACOMS inbound message, and an XML-formatted validation report. The documents are routed based on the success or failure of transformation.

The outbound application channel creates a TRADACOMS message from XML and an XML-formatted validation report. The documents are routed based on the success or failure of transformation.

TRADACOMS Inbound and Outbound Applications Prerequisites

Before you continue, ensure that the following prerequisites are met:

- You have a working knowledge of iSM and iIT.

- iSM Version 8.0 is installed.
- iWay TRADACOMS Adapter is installed.
- iIT Version 8.0 is installed.
- System and channel Special Registers (SREGs) are updated to match your directory structure, as shown in [Extract User Samples for TRADACOMS](#).

Extracting TRADACOMS User Samples

This section describes how to extract user samples for TRADACOMS.

Extract User Samples for TRADACOMS

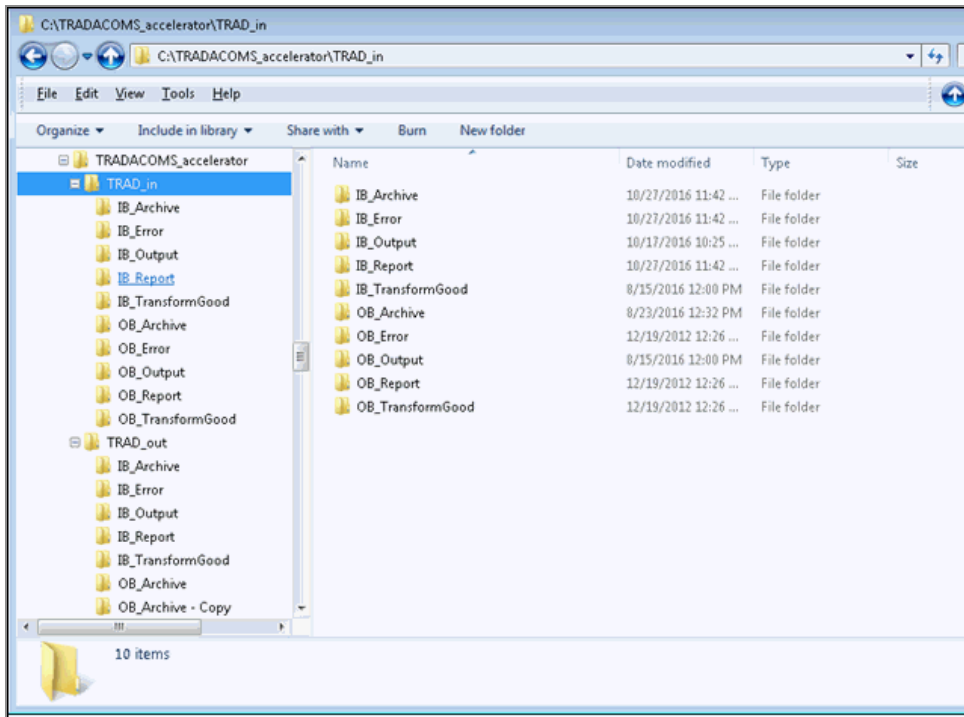
Procedure

1. Download the TRAD_usr_samples.zip file containing the TRADACOMS user sample workspace from the following website:

<http://techsupport.informationbuilders.com>

The downloaded TRADACOMS_usr_samples.zip contains the following files:

- TRADACOMS_Accelerator.zip
 - TRAD_usr_samples_iIT_workspace.zip
2. Save the TRADACOMS_usr_samples_iIT_workspace.zip file to a folder on your local drive.
 3. Unzip the TRADACOMS_Accelerator.zip file to the location where you want to store your data, as shown in the following image.

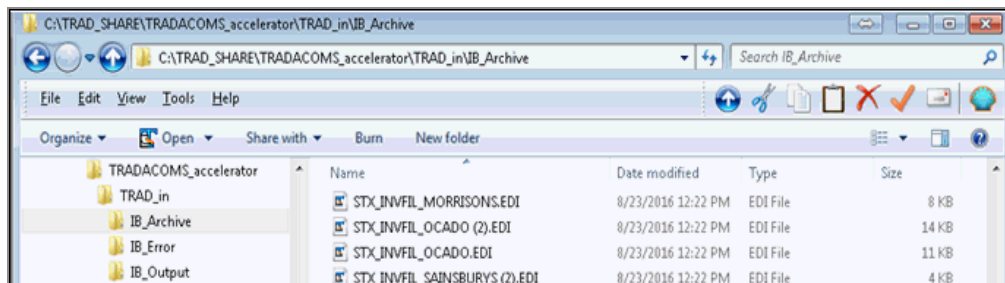


The TRADACOMS_Accelerator.zip file contains sample input and output data that you can use.

- Inbound test data is located in the following folder:

\TRADACOMS_Accelerator\TRAD_in\IB_Archive

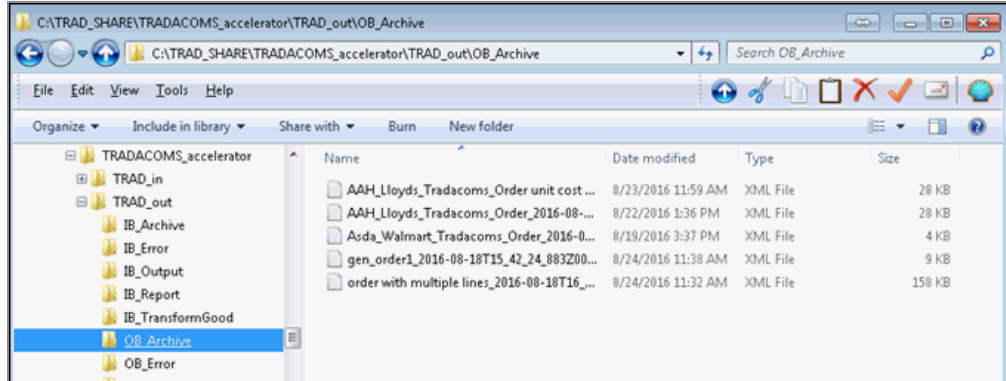
For example:



- Outbound test data is located in the following folder:

\TRADACOMS_Accelerator\TRAD_out\OB_Archive

For example:



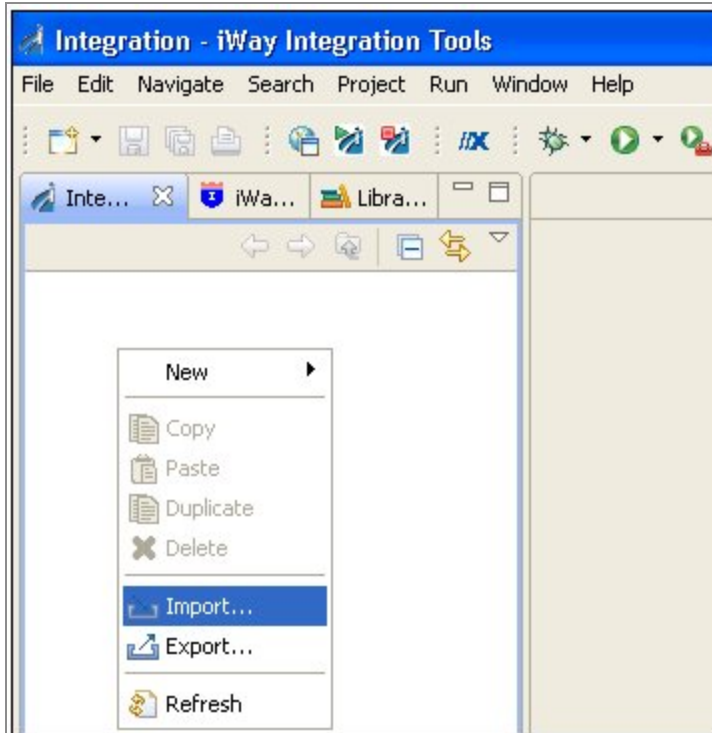
Importing TRADACOMS User Samples to iIT as a Workspace

This section describes how to import TRADACOMS user samples to iWay Integration Tools (iIT) as a workspace.

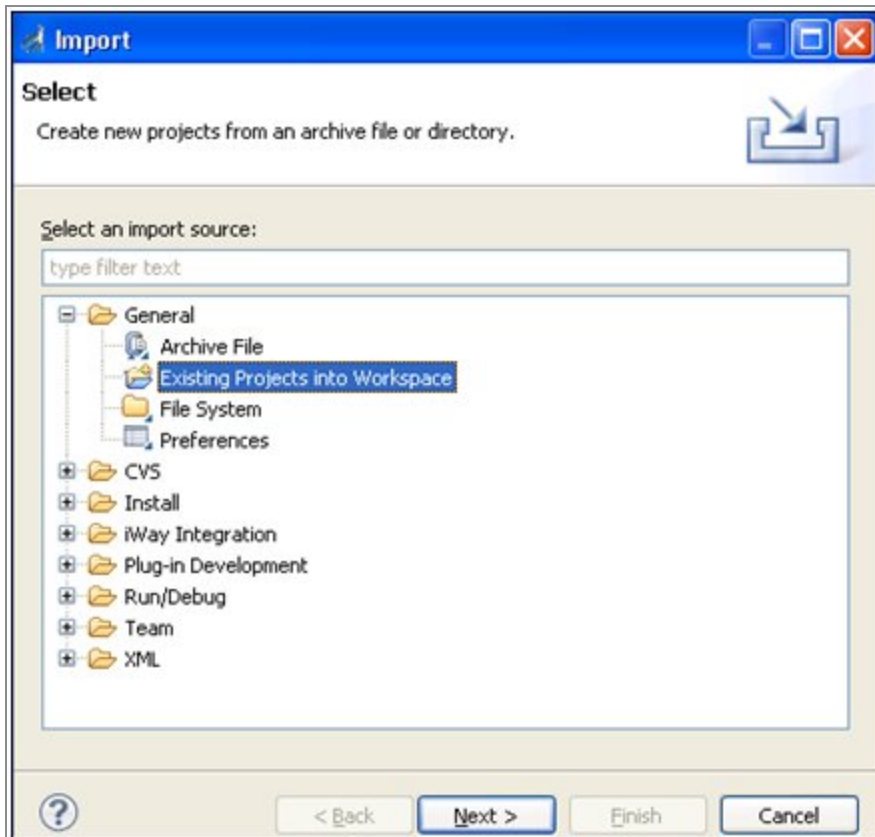
Import TRADACOMS User Samples to iIT as a Workspace

Procedure

1. Start iIT.
2. Right-click anywhere inside the Integration Explorer tab and select **Import...** from the context menu.

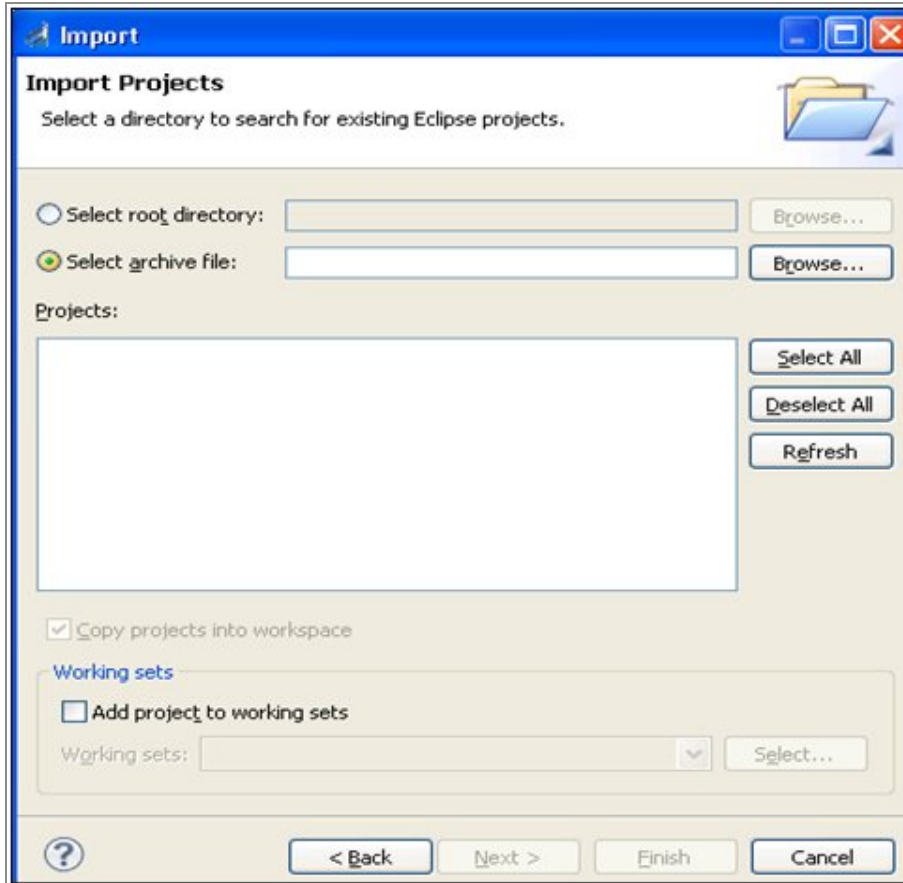


The Import dialog opens.



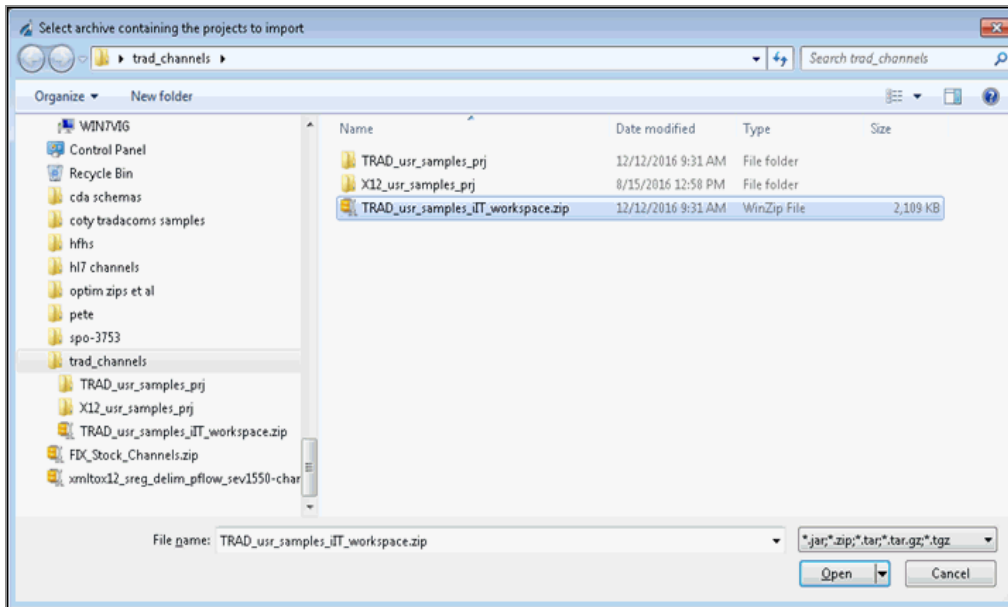
3. Expand the **General** folder, select **Existing Projects into Workspace**, and then click **Next**.

The Import Projects pane opens, as shown in the following image.



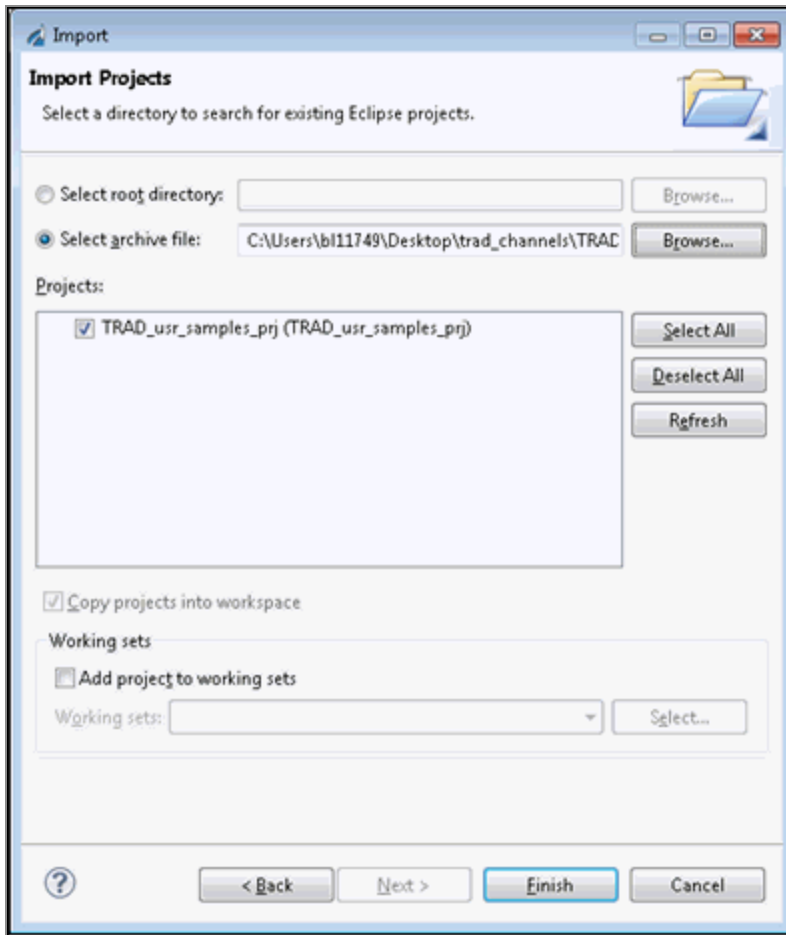
4. Click **Select archive file** and then click **Browse**.

The Select archive containing the projects to import pane opens.



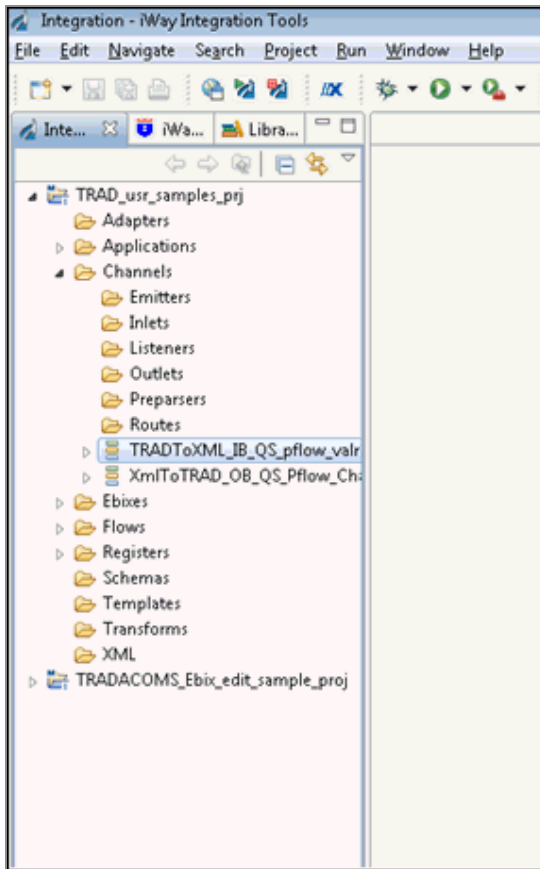
5. Select the *TRAD_usr_samples_iIT_workspace.zip* file and click **Open**.

You are returned to the Import Projects pane, as shown in the following image.



6. Click **Finish**.

The TRADACOMS user samples are loaded into your iIT workspace, as shown in the following image.



The Integration Explorer tab on the left pane displays a hierarchy of all the imported channel components (for example, Ebixes, listeners, outlets, preparers, routes, process flows, and so on).

Publishing iIAs to the iSM Registry

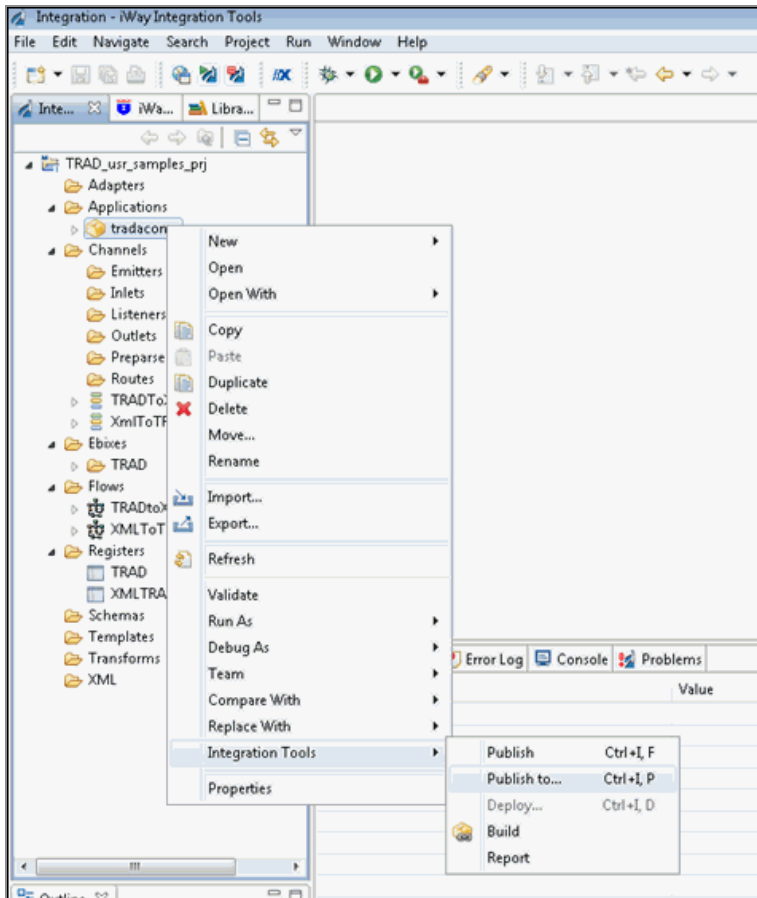
This section describes how to publish iWay Integration Applications (iIAs) to the iWay Service Manager (iSM) Registry.

Publish iIAs to the iSM Registry

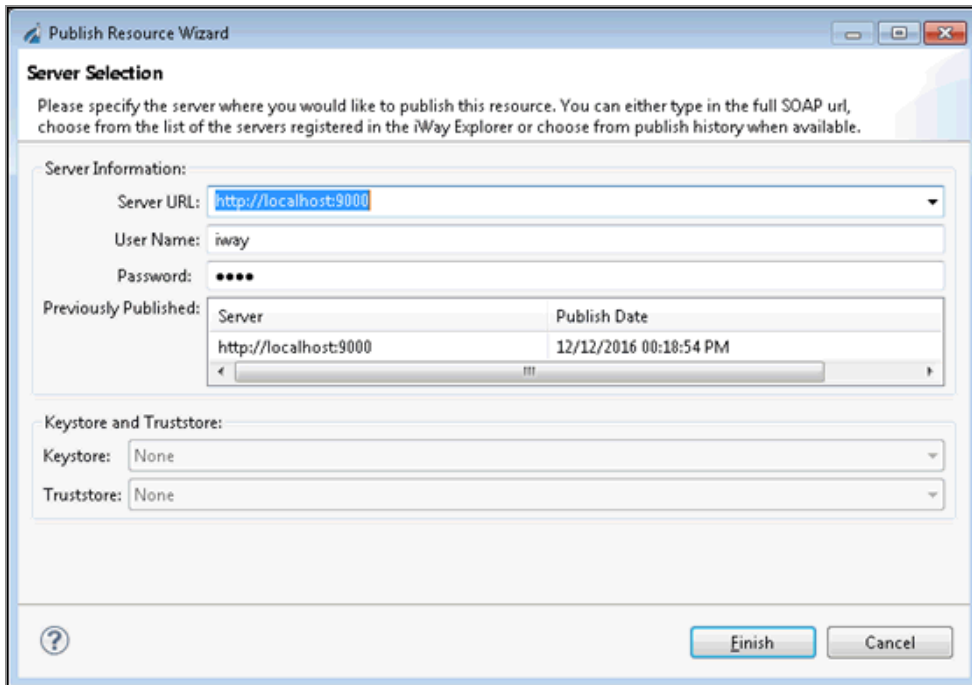
Procedure

1. In the Integration Explorer tab, right-click the application name, select **Integration**

Tools from the context menu, and then click **Publish to...**, as shown in the following image.



The Publish Resource Wizard dialog opens.



- In the Server URL field, type the server IP number or computer name and then the port number (default port is 9000). For example:

```
http://111.111.111.000:9000
```

Type the iSM credentials (for example, user name: `iway`, password: `iway`).

- Click **Finish**.

The Console tab on the bottom provides a status log that you can use for verification purposes, as shown in the following image.



Deploying iWay Integration Applications to iWay Service Manager

This section describes how to deploy iWay Integration Applications (iIAs) to iWay Service Manager (iSM).

Deploy iWay Integration Applications to iWay Service Manager

Procedure

1. Enter the following URL to access the iSM Administration Console:

```
http://[host]:[port]/ism
```

where:

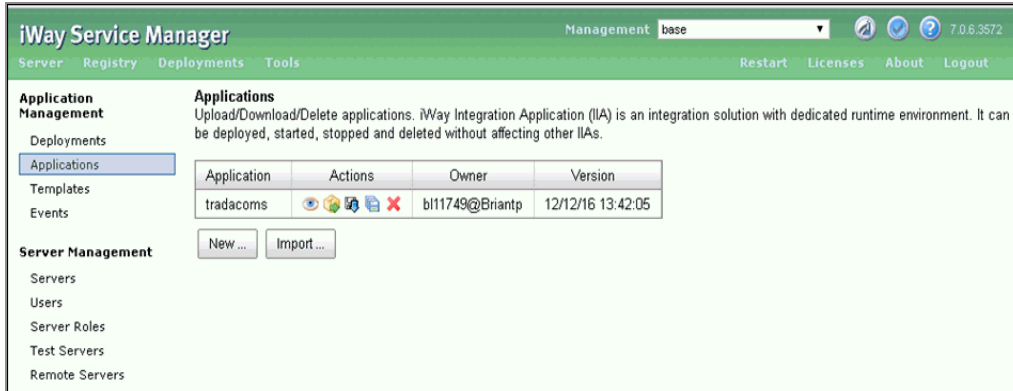
host


Is the host machine where iSM is installed. The default value is *localhost*.

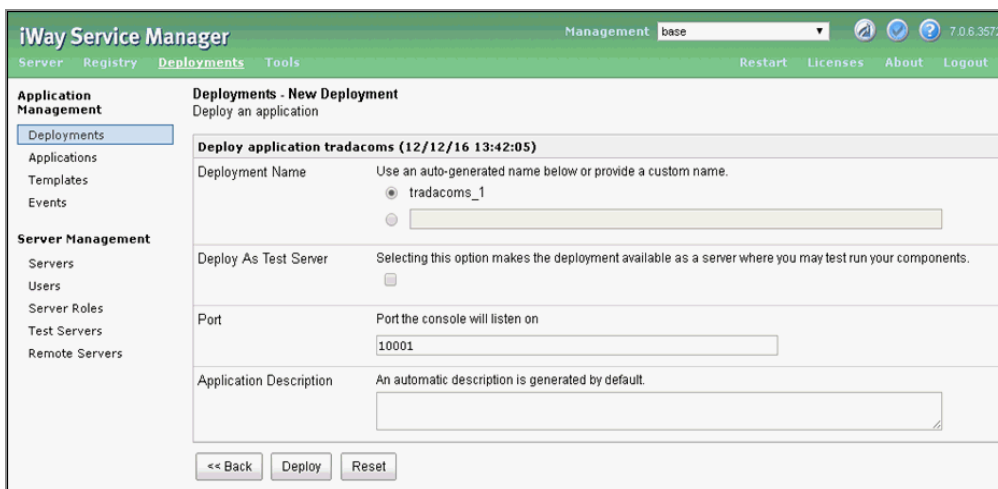
port

Is the port where iSM is listening. The default port is *9999*.

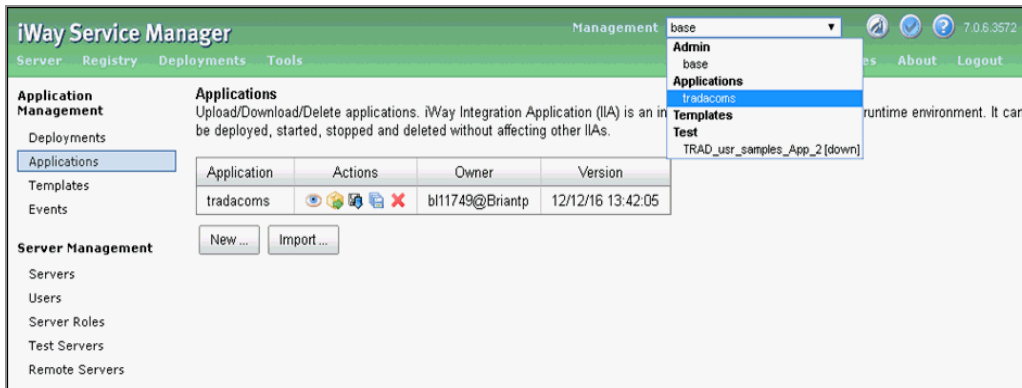
2. After publishing the iIA, you can find it under the Management\Applications link in the iSM Administration Console, as shown in the following image.



3. Click the **Deploy** icon  next to the application name under the Actions column.
The Deployments pane opens, as shown in the following image.




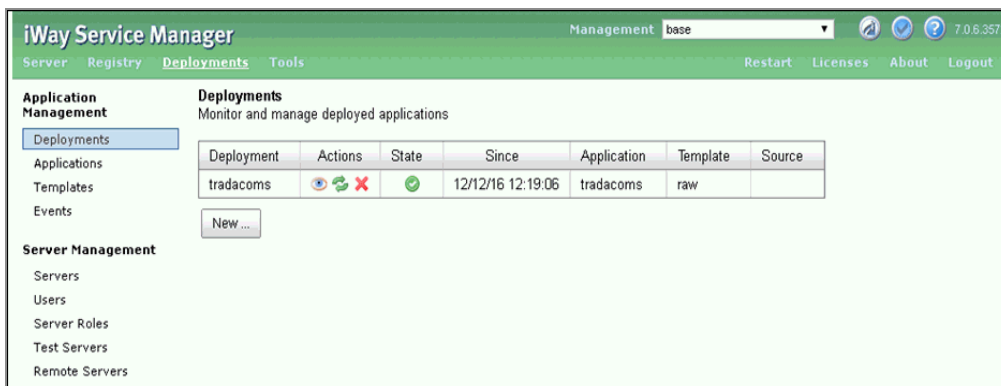
4. Click **Deploy**.
5. From the Management dropdown list, select your deployed application.



- Click **Server** in the top menu and then **Register Settings** in the left pane for the application.
- Click **Add** to create all required registers (*TRAD_Installdir*, *TRAD_Input*, *TRAD_Output*, and *ValidateTRAD*) for the application.

For more information, see BAD XREF HERE "Setting Registers in the iWay Service Manager Administration Console.

- In the State column, click the **Deployment State**  icon to start the deployed Application.



- When the *Are you sure you want to start* message appears, click **OK** to proceed.
- Once the application has successfully started, place your input data into the input location that is configured for the application.
- Select the application from the Management dropdown list.
- Click the **Monitoring** link. The deployed application channels within the iIA are displayed, as shown in the following image.

Name	Type	State	Waiting	Messages					Description
				Active	Completed	Successful	Failed	Since Last Refresh	
TRADToXML_IB_QS_pflow_valrpt_channel.TRADToXML	FILE	✔	NA						TRADACOMS to XML file listener
XmiToTRAD_OB_QS_Pflow_Channel2.inlet.1:XmiToTRAC	FILE	✔	NA						Listeners are protocol handlers, that receive input for a channel from a configured endpoint. Listed below are references to the listeners that are defined in the registry.

You can stop either channel and have only one channel running at a time as required.

Configuring Special Registers

This section describes how to configure the required Special Registers (SREGs) for the iWay Integration Solution for TRADACOMS using the iWay Service Manager (iSM) Administration Console. You must configure the following SREGs:

- TRAD_Installdir
- TRAD_Input
- TRAD_Output

These SREGs will be used by the pre-configured channels for inbound and outbound processing to route XML and TRADACOMS messages as required.

Configure Special Registers

Procedure

1. Create the following new folder on your file system:

```
C:\TRADACOMS_Accelerator
```

2. Log on to the iSM Administration Console.

- Click **Server** in the menu bar located on the top of the iSM Administration Console and then click **Register Settings** in the left pane.

The Register Settings pane opens, as shown in the following image.

iWay Service Manager Management base 7.0.6.3572

Server Registry Deployments Tools Restart Licenses About Logout

Properties
General Properties
Java Properties

Settings
General Settings
Console Settings
Java Settings
Register Settings
Trace Settings
Log Settings
Path Settings
Data Settings
Backup Settings

Providers
Data Provider
Services Provider
Directory Provider
Security Provider
XML Namespace Map Provider
Pooling Providers
Authentication Realms
Data Quality Providers
TCP Connection Provider
Schedule Provider
Calendar Provider
SNMP Provider
Secure Shell Provider

Facilities
Activity Facility
Correlation Facility
Command Consoles

Register Settings
Special registers are named variables that reference values which are carried throughout the system. Once defined, these variables become available to all components of the system. Any changes to the register settings do not take effect until the server is restarted/redeployed. Listed below are the register settings for the base configuration of this server.

Special Registers

<input type="checkbox"/>	Name	Value	Description	Type
<input type="checkbox"/>	iwayversion	7.0.6	system defined (readonly)	string
<input type="checkbox"/>	iwayhome	C:/PROGRA~2/iway7/	system defined (readonly)	string
<input type="checkbox"/>	iwaydata	C:/PROGRA~2/iway7/	system defined (readonly)	string
<input type="checkbox"/>	iway.startup.time	1478815813720	system defined (readonly)	string
<input type="checkbox"/>	iway.config	base	system defined (readonly)	string
<input type="checkbox"/>	engine	base	system defined (readonly)	string
<input type="checkbox"/>	iwayconfig	base	system defined (readonly)	string
<input type="checkbox"/>	iwayworkdir	C:/PROGRA~2/iway7/config/base	system defined (readonly)	string
<input type="checkbox"/>	iway.workdir	C:/PROGRA~2/iway7/config/base	system defined (readonly)	string
<input type="checkbox"/>	iway.serverip	172.30.234.118	system defined (readonly)	string
<input type="checkbox"/>	iway.serverhost	INFORMA-Q8T67IU	system defined (readonly)	string
<input type="checkbox"/>	iway.serverfullhost	INFORMA-Q8T67IU.ibi.com	system defined (readonly)	string
<input type="checkbox"/>	iway.pid	1956	system defined (readonly)	string
<input type="checkbox"/>	jce.unlimited	false	system defined (readonly)	string
<input type="checkbox"/>	envoy-port	9001		integer
<input type="checkbox"/>	ibse-port	9000		integer
<input type="checkbox"/>	sentinel-pwd	[not viewable]		password
<input type="checkbox"/>	sentinel-uid	admin		string
<input type="checkbox"/>	sentinel-url	http://localhost:8080		string

Add **Delete**

- Click **Add**.

The Register Settings pane opens, as shown in the following image.

Register Settings
Special registers are named variables that reference values which are carried throughout the system. Once defined, these variables become available to all components of the system. Any changes to the register settings do not take effect until the server is restarted/redeployed. Listed below are the register settings for the base configuration of this server.

Special Register Definition

Name *	Enter the name of the special register to add. <input type="text" value="TRAD_Installdir"/>
Type	Select a type for the value of this special register. <input type="text" value="string"/>
Value *	Enter a value for this special register. The value can be a constant or a call to the evaluation functions. <input type="text" value="C:\TRADACOMS_Accelerator"/>
Description	Enter a description for this special register. <input type="text"/>

5. Perform the following steps:

- a. Type the following in the Name field:

```
TRAD_Installdir
```

- b. Type the following in the Value field, which is the new folder you created on your file system in Step 1:

```
C:\TRADACOMS_Accelerator
```

- c. Click **Finish**.

You are returned to the Special Registers pane, where the new SREG you created (TRAD_Installdir) is now listed, as shown in the following image.

Special Registers				
<input type="checkbox"/>	Name	Value	Description	Type
<input type="checkbox"/>	iwaversion	7.0.6	system defined (readonly)	string
<input type="checkbox"/>	iwayhome	C:/PROGRA~2/iway7/	system defined (readonly)	string
<input type="checkbox"/>	iwaydata	C:/PROGRA~2/iway7/	system defined (readonly)	string
<input type="checkbox"/>	iway.startup.time	1478815813720	system defined (readonly)	string
<input type="checkbox"/>	iway.config	base	system defined (readonly)	string
<input type="checkbox"/>	engine	base	system defined (readonly)	string
<input type="checkbox"/>	iwayconfig	base	system defined (readonly)	string
<input type="checkbox"/>	iwayworkdir	C:/PROGRA~2/iway7/config/base	system defined (readonly)	string
<input type="checkbox"/>	iway.workdir	C:/PROGRA~2/iway7/config/base	system defined (readonly)	string
<input type="checkbox"/>	iway.serverip	172.30.234.118	system defined (readonly)	string
<input type="checkbox"/>	iway.serverhost	INFORMA-Q8T67IU	system defined (readonly)	string
<input type="checkbox"/>	iway.serverfullhost	INFORMA-Q8T67IU.ibi.com	system defined (readonly)	string
<input type="checkbox"/>	iway.pid	1956	system defined (readonly)	string
<input type="checkbox"/>	jce.unlimited	false	system defined (readonly)	string
<input type="checkbox"/>	TRAD_Installdir	C:\TRADACOMS_Accelerator		string
<input type="checkbox"/>	envoy-port	9001		integer
<input type="checkbox"/>	lbse-port	9000		integer
<input type="checkbox"/>	sentinel-pwd	[not viewable]		password
<input type="checkbox"/>	sentinel-uid	admin		string
<input type="checkbox"/>	sentinel-url	http://localhost:8080		string

6. Click **Add**.

The Register Settings pane opens, as shown in the following image.

Register Settings
 Special registers are named variables that reference values which are carried throughout the system. Once defined, these variables become available to all components of the system. Any changes to the register settings do not take effect until the server is restarted/redeployed. Listed below are the register settings for the base configuration of this server.

Special Register Definition

Name * Enter the name of the special register to add.

Type Select a type for the value of this special register.

Value * Enter a value for this special register. The value can be a constant or a call to the evaluation functions.

Description Enter a description for this special register.

7. Perform the following steps:

- a. Type the following in the Name field:

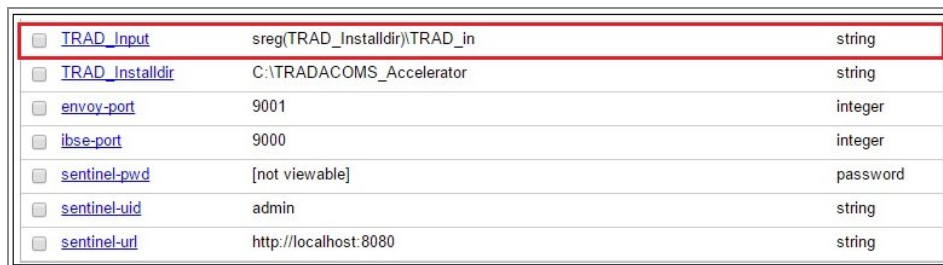
```
TRAD_Input
```

- b. Type the following in the Value field, which is the input folder that will be used:

```
sreg(TRAD_Installdir)\TRAD_in
```

- c. Click
- Finish**
- .

You are returned to the Special Registers pane, where the new SREG you created (TRAD_Input) is now listed, as shown in the following image.



<input type="checkbox"/> TRAD_Input	sreg(TRAD_Installdir)\TRAD_in	string
<input type="checkbox"/> TRAD_Installdir	C:\TRADACOMS_Accelerator	string
<input type="checkbox"/> envoy-port	9001	integer
<input type="checkbox"/> ibse-port	9000	integer
<input type="checkbox"/> sentinel-pwd	[not viewable]	password
<input type="checkbox"/> sentinel-uid	admin	string
<input type="checkbox"/> sentinel-url	http://localhost:8080	string

8. Click
- Add**
- .

The Register Settings pane opens, as shown in the following image.

Register Settings
Special registers are named variables that reference values which are carried throughout the system. Once defined, these variables become available to all components of the system. Any changes to the register settings do not take effect until the server is restarted/redeployed. Listed below are the register settings for the base configuration of this server.

Special Register Definition

Name * Enter the name of the special register to add.

Type Select a type for the value of this special register.

Value * Enter a value for this special register. The value can be a constant or a call to the evaluation functions.

Description Enter a description for this special register.

Finish

9. Perform the following steps:

- a. Type the following in the Name field:

```
TRAD_Output
```

- b. Type the following in the Value field, which is the output folder that will be used:

```
sreg(TRAD_Installdir)\TRAD_out
```

- c. Click
- Finish**
- .

You are returned to the Special Registers pane, where the new SREG you created (TRAD_Output) is now listed, as shown in the following image.


<input type="checkbox"/>	TRAD_Input	sreg(TRAD_Installdir)\TRAD_in	string
<input type="checkbox"/>	TRAD_Installdir	C:\TRADACOMS_Accelerator	string
<input type="checkbox"/>	TRAD_Output	sreg(TRAD_Installdir)\TRAD_out	string
<input type="checkbox"/>	envoy-port	9001	integer
<input type="checkbox"/>	ibse-port	9000	integer
<input type="checkbox"/>	sentinel-pwd	[not viewable]	password
<input type="checkbox"/>	sentinel-uid	admin	string
<input type="checkbox"/>	sentinel-url	http://localhost:8080	string

Stopping Inbound and Outbound Processing

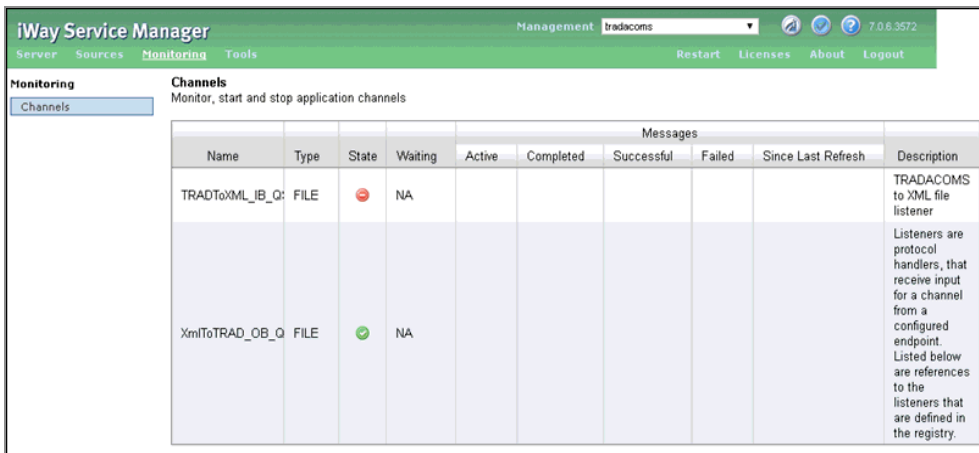
This section describes how to stop inbound (TRADACOMS to XML) and outbound (XML to TRADACOMS) processing.



Stop Inbound (TRADACOMS to XML) Processing

Procedure

1. Click the **State** icon  adjacent to the inbound application channel under Management\Monitoring and click **OK**.


The inbound application channel will be stopped, as shown in the following image.



iWay Service Manager											
Management tradacoms											
Server Sources Monitoring Tools											
Restart Licenses About Logout											
Monitoring											
Channels											
Monitor, start and stop application channels											
Name	Type	State	Waiting	Messages							Description
				Active	Completed	Successful	Failed	Since Last Refresh			
TRADToXML_IB_Q	FILE		NA								TRADACOMS to XML file listener
XmlToTRAD_OB_Q	FILE		NA								Listeners are protocol handlers, that receive input for a channel from a configured endpoint. Listed below are references to the listeners that are defined in the registry.

Stop Outbound (XML to TRADACOMS) Processing

Procedure

1. Click the **State** icon  adjacent to the outbound application channel under Management\Monitoring and click **OK**.

The outbound application channel will be stopped, as shown in the following image.

Name	Type	State	Waiting	Active	Completed	Messages		Since Last Refresh	Description
						Successful	Failed		
TRADToXML_IB_Q	FILE	✔	NA						TRADACOMS to XML file listener
XmlToTRAD_OB_Q	FILE	✖	NA						Listeners are protocol handlers, that receive input for a channel from a configured endpoint. Listed below are references to the listeners that are defined in the registry.

Testing Sample Applications

This section describes how to test the sample inbound (TRADACOMS to XML) and outbound (XML to TRADACOMS) applications.

Test the Sample Inbound (TRADACOMS to XML) Application

Procedure

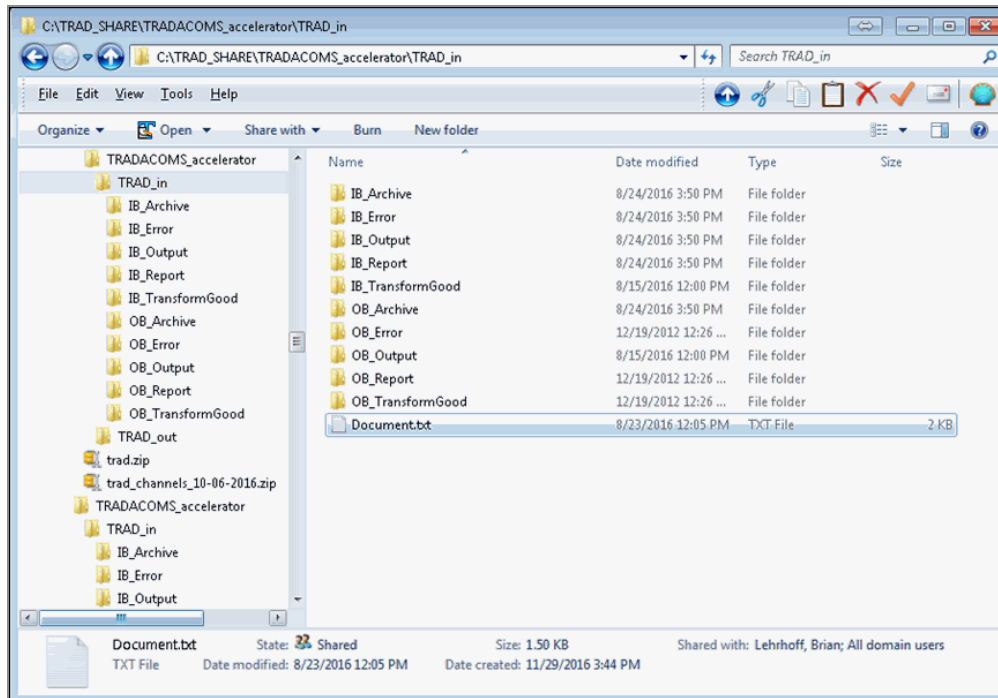
1. Copy a test data file from the following directory:

```
TRADACOMS_Accelerator\TRAD_IN\IB_Archive
```

to:

```
TRADACOMS_Accelerator\TRAD_in
```

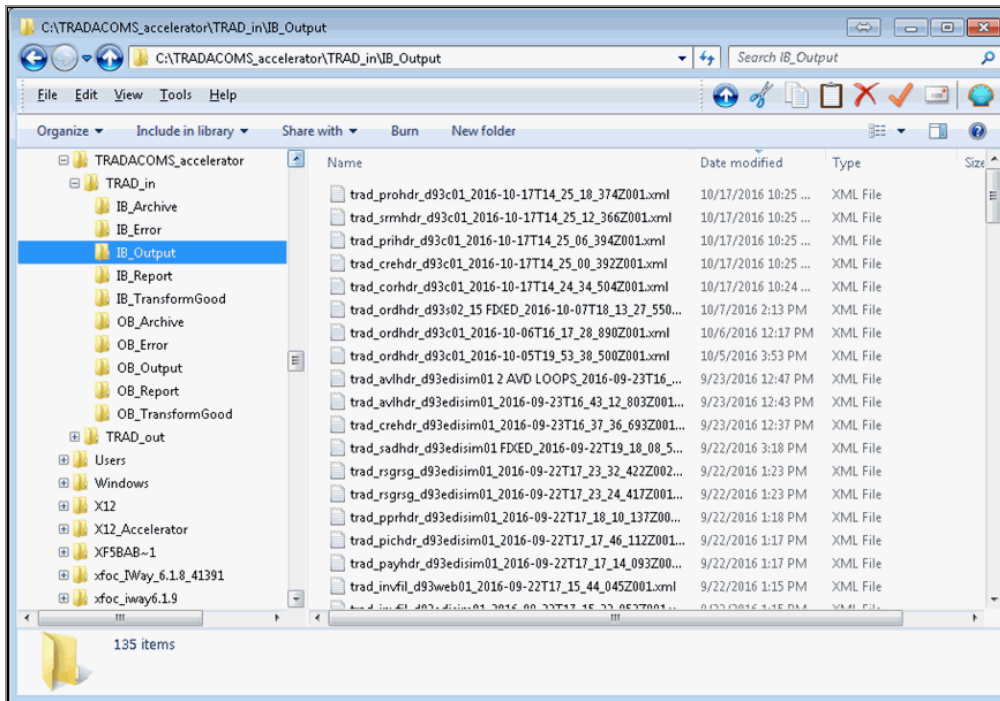
For example:



2. Observe the transformed XML output in the following directory:

TRADACOMS_Accelerator\TRAD_in\IB_Output

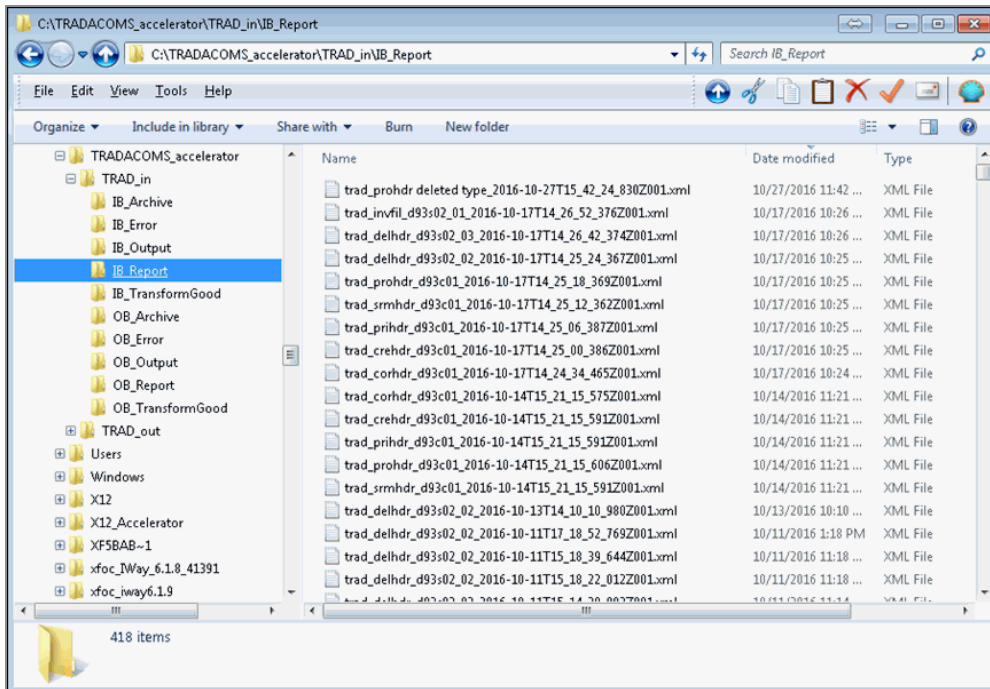
For example:



3. Observe the validation report in the following directory:

TRADACOMS_Accelerator\TRAD_in\IB_Report

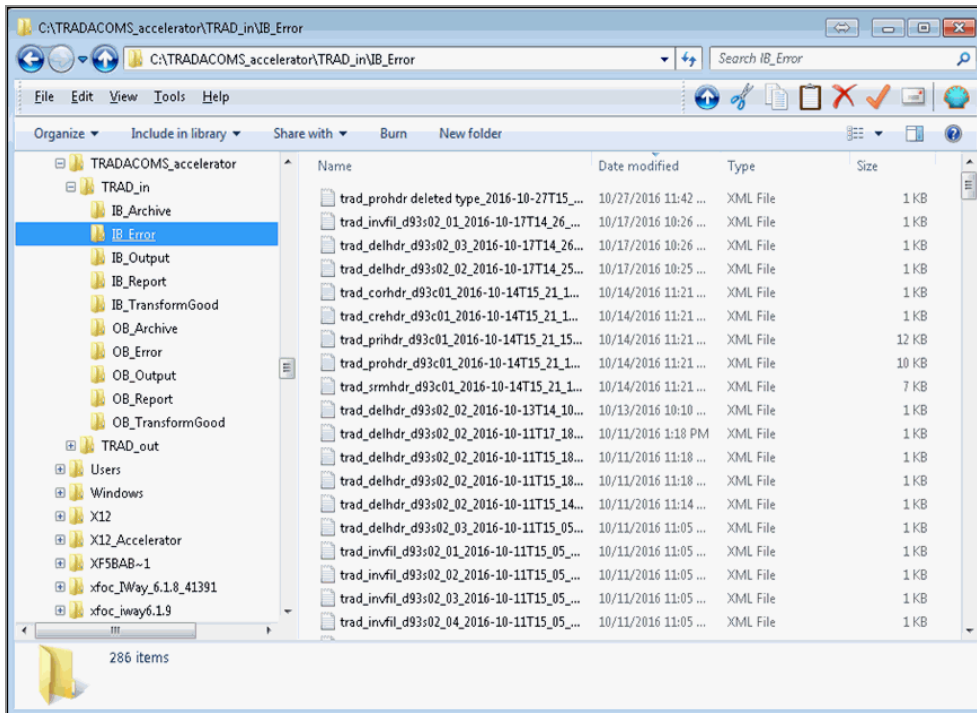
For example:



4. If any Errors occur, then observe the following directory:

TRADACOMS_Accelerator\TRAD_in\IB_Error

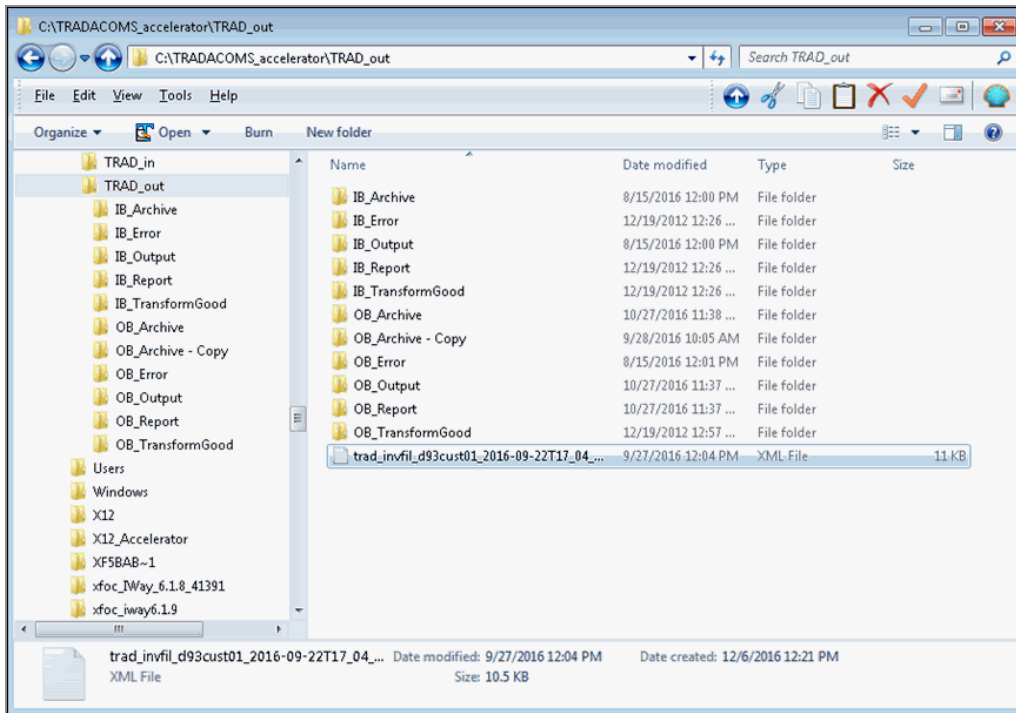
For example:



5. After transformation, the input file will get stored in the following directory:

TRADACOMS_Accelerator\TRAD_in\IB_Archive

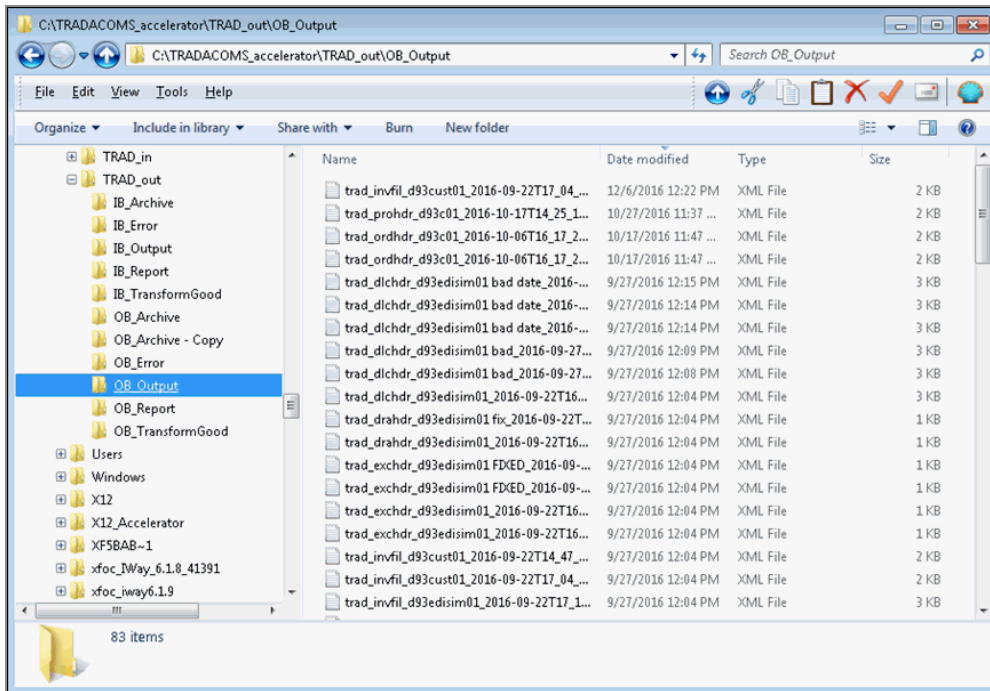
For example:



2. Observe the TRADACOMS formatted output in the following directory:

TRADACOMS_Accelerator\TRAD_out\OB_Output

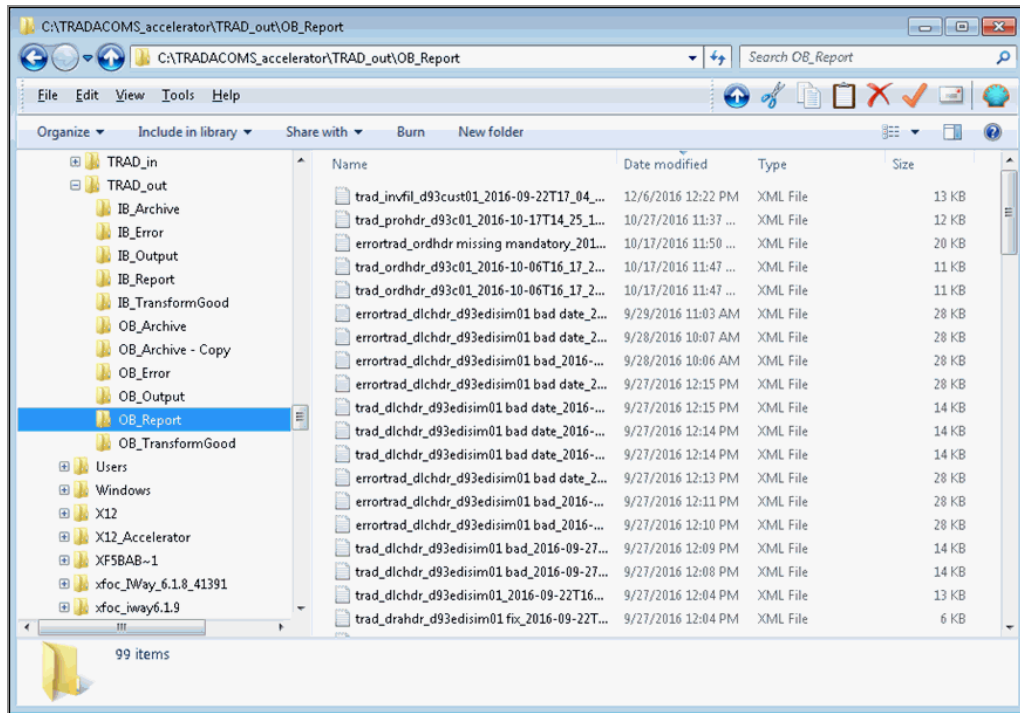
For example:



3. Observe the validation report in the following directory:

TRADACOMS_Accelerator\TRAD_out\OB_Report

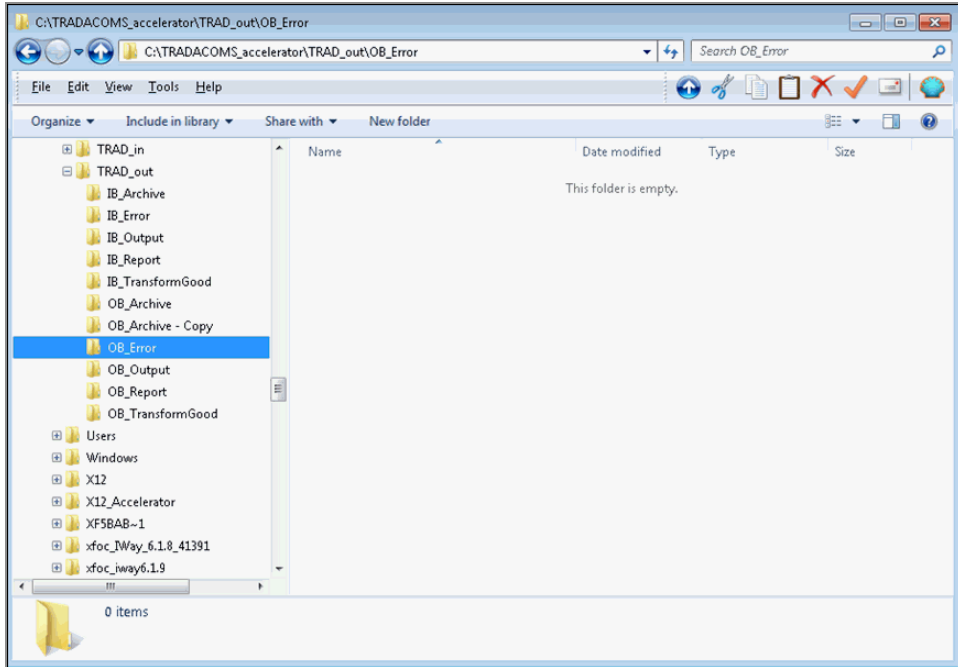
For example:



4. If any Errors occur, then observe the following directory:

TRADACOMS_Accelerator\TRAD_out\OB_Error

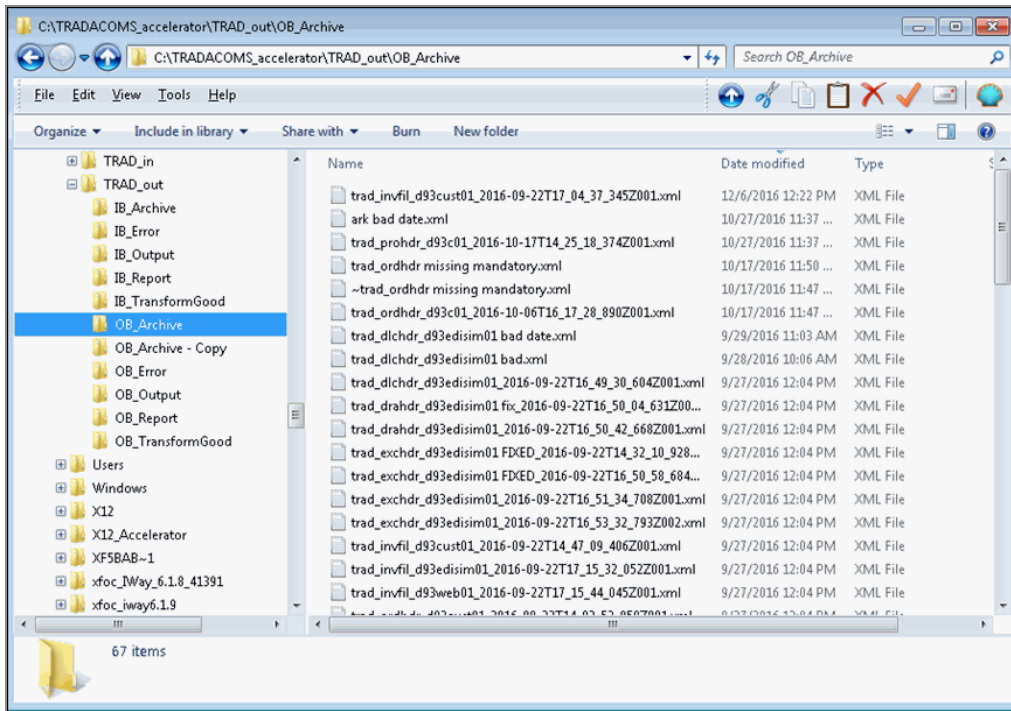
For example:



5. After transformation, the input XML file will get stored in the following directory:

TRADACOMS_Accelerator\TRAD_out\OB_Archive

For example:



Inbound Processing: TRADACOMS to XML

The iWay Integration Solution for TRADACOMS runs within iWay Service Manager (iSM). iSM converts a document from TRADACOMS format to XML format, and validates it based on TRADACOMS published implementation guides.

This chapter provides the information you need to understand and implement a basic inbound message flow.

Inbound Processing Overview

This section describes the iWay business components and the processing steps in the basic inbound message flow.

The inbound process converts a TRADACOMS formatted document to an XML document. Inbound processing consists of the following components in the sample message flow:

- **Inlet**

- The file **listener** picks up the incoming TRADACOMS document.
- The **preparser** obtains the message type from the TRADACOMS document. The preparser converts the original TRADACOMS document to an XML representation of that document.

The iWay Integration Solution for TRADACOMS supports one preparser, XDTRADPreParser (com.ibi.preparsers.XDTRADPreParser), which is provided by iWay Software.

- **Validation**

The inbound TRADACOMS document is validated for structure and content. The published TRADACOMS standards and user implementation guides define element types (for example, numeric, alpha, or date) and describe business rules to apply for validation.

For example, the following is a typical date segment in an inbound TRADACOMS document:

```
FIL=7012+1758+160901+FL RLX'
```

Note: All TRADACOMS dates use a six digit date format (YYMMDD).

- **Route and Process Flow**

In our sample message flow, the validation report service (TRAD_Validation_rpt) creates a validation report in XML format. This report indicates success or failure transformation and contains the input and output documents. The route directs the document based on this report. You can also apply additional business logic by adding a service to the flow.

For more information on the services available with iSM, see the *ibi™ iWay® Service Manager Component Reference Guide*.

- **Outlet**

Outlets define how messages leave a channel at the end of a process. In this example, a *Passthrough* emitter is used. All the file emits are done in the process flow.

For more information on the emitters available with iSM, see the *ibi™ iWay® Service Manager User Guide*.

For more information on deploying and starting the sample channels within the iWay Integration Application (iIA), see [Working With TRADACOMS Inbound and Outbound Applications Using iWay Integration Tools](#).

Special Register Sets

A Special Register (SREG) is a variable that users can set. As a best practice within the EDI framework, you can use SREGs to define input and output locations. When an iWay Integration Application (iIA) containing EDI channels is migrated between systems (for example, from *Dev* to *UAT*), the only changes required to deploy the iIA in the new system is to set the SREG and start the iIA. EDI channels use multiple locations and paths. This practice minimizes the migration effort. For a list of SREGs that are provided with iSM, see the *ibi™ iWay® Service Manager Programmer's Guide*. For more information on defining an SREG of your own, see the *ibi™ iWay® Service Manager User Guide*.

The sample inbound TRADACOMS channel sets all of the SREGs that are listed in the following table.

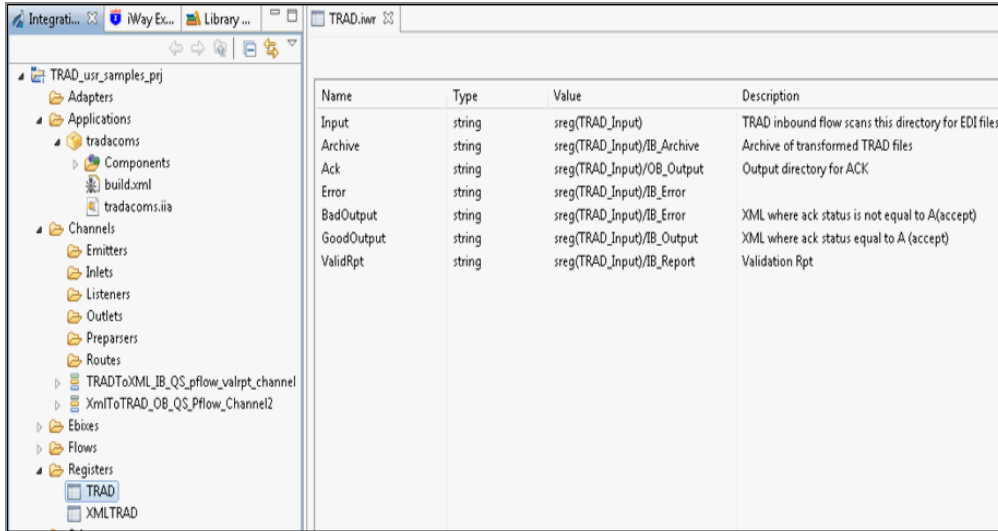
Name	Value
Input	sreg(TRAD_INPUT)
Archive	sreg(TRAD_INPUT)/IB_Archive
Ack	sreg(TRAD_INPUT)/OB_Output
Error	sreg(TRAD_INPUT)/IB_Error
BadOutput	sreg(TRAD_INPUT)/IB_Error
GoodOutput	sreg(TRAD_INPUT)/IB_Output
ValidRpt	sreg(TRAD_INPUT)/IB_Report

Modify SREG Sets in the Channel

Procedure

1. In the Integration Explorer tab (left pane of iIT), double-click the name of the SREG set (for example, TRAD).

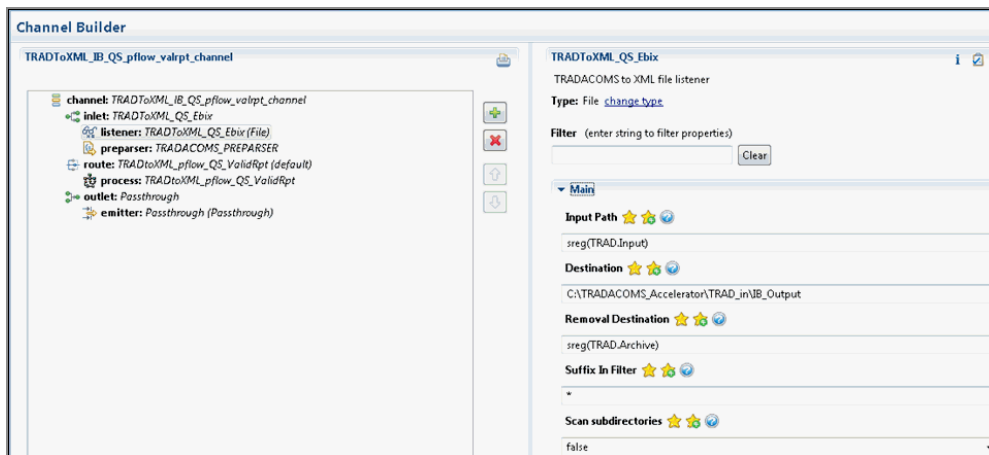
The selected SREG set opens as a new tab, as shown in the following image.



2. Ensure to save your changes.
3. If you make any changes to the SREG set, rebuild and redeploy the iIA for the changes to take effect.

Sample Channel File Listener

The following image shows a sample File listener that has been configured using the Channel Builder in iIT.



The following table describes the EDI-specific configuration parameters for the File listener. An asterisk (*) indicates that a parameter is required. For parameters not listed in the following table, accept the default value. Ensure that you have created any directories that are referenced by the SREGs, or errors will occur during deployment.

Parameter	Value
Input Path *	sreg(TRAD.Input) This value is an SREG that uses a defined directory in which input messages are received.
Destination *	sreg(TRAD.ListenerOutput) This value is an SREG that uses a defined directory in which output files are stored after transformation.
Removal Destination	sreg(TRAD.Archive) This value is an SREG that uses a defined directory to which input messages are moved if they fail during transformation. It is recommended to configure a removal destination when you are constructing a basic channel.
Suffix In	* Input files with any file extension are allowed.
Suffix Out	XML The extension for output files is .xml .

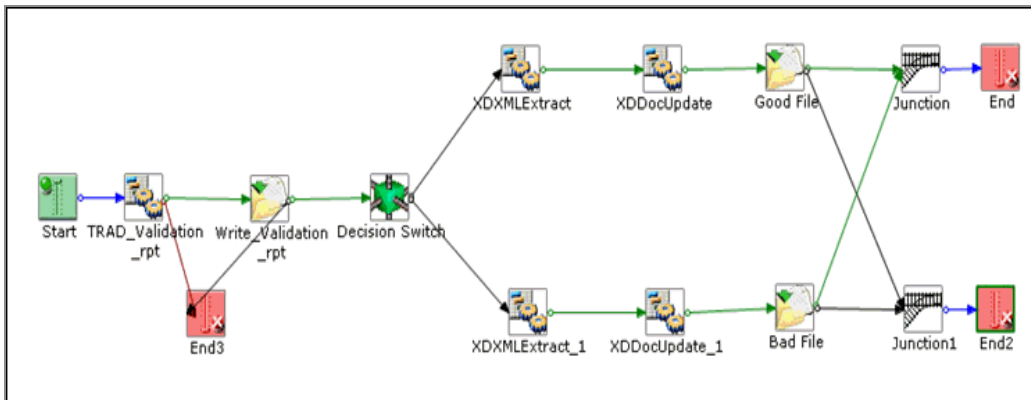
Sample Channel Preparser

The TRADACOMS preparser used in the sample channel does not have any parameters that can be modified by the user.

Sample Channel Process Flow

A route is defined in the sample channel that invokes the *TRADtoXML_pflow_QS_validreport* process flow. Using the validation report, valid transformed XML data is output to one folder and invalid transformed data will be written to another folder. If transformation in the preparer fails, then a *bad* validation report will be written. Bad validation reports can be routed through email or a queue for further inspection and remediation.

The following image shows the process flow from the sample channel.



Prior versions of the iWay e-Business user documentation demonstrate how to build channels and process flows in more detail using the iSM Administration Console and iIT.

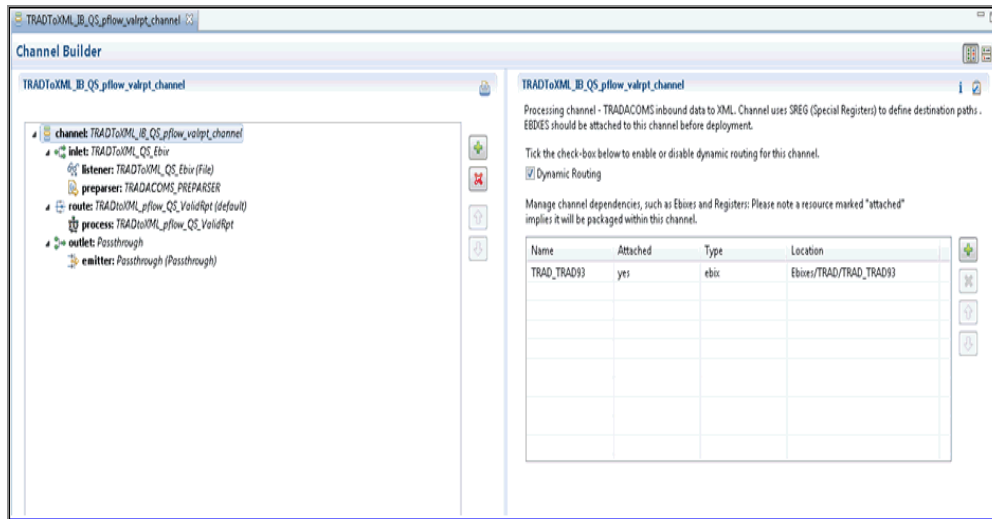
Adding an Ebix to a Channel

This section describes how to add an Ebix to the channel.

Add an Ebix to a Channel

Procedure

1. Open Channel Builder in iIT and then select the name of the channel in the upper-left pane.



2. Click the green plus (+) symbol to add a new Ebix. You can also select an Ebix and click the red (x) symbol to delete an Ebix.

If your channel contains multiple Ebixes, then use the up and down arrows to order the Ebixes. Ebixes are executed in the order that they appear on this screen, from top to bottom.

3. Ensure to save your changes.
4. If you make any changes to the channel, rebuild and redeploy the iIA for the changes to take effect.

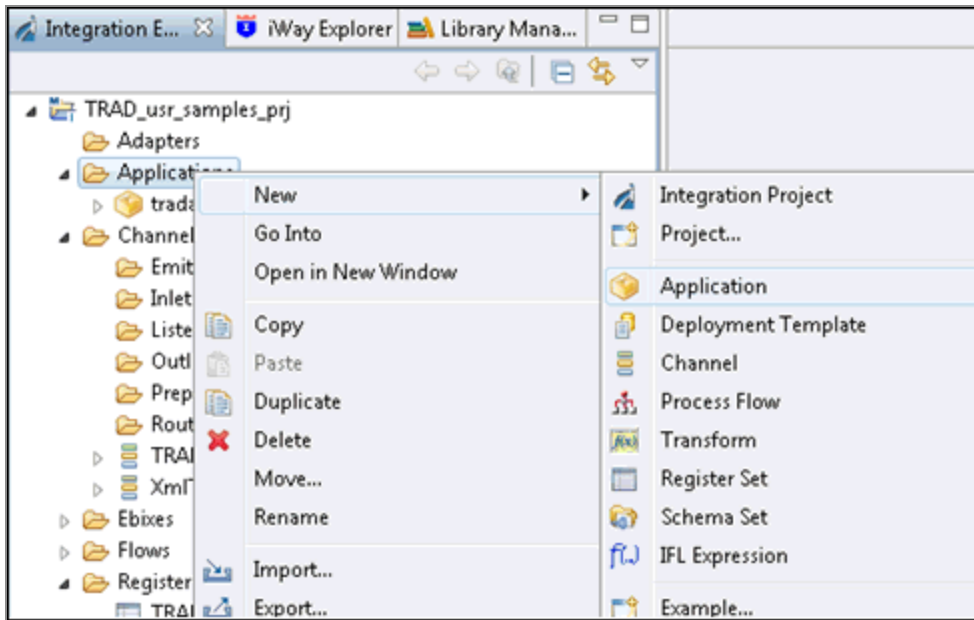
Rebuilding Your Application

If any changes are made to your application, then you must rebuild your application.

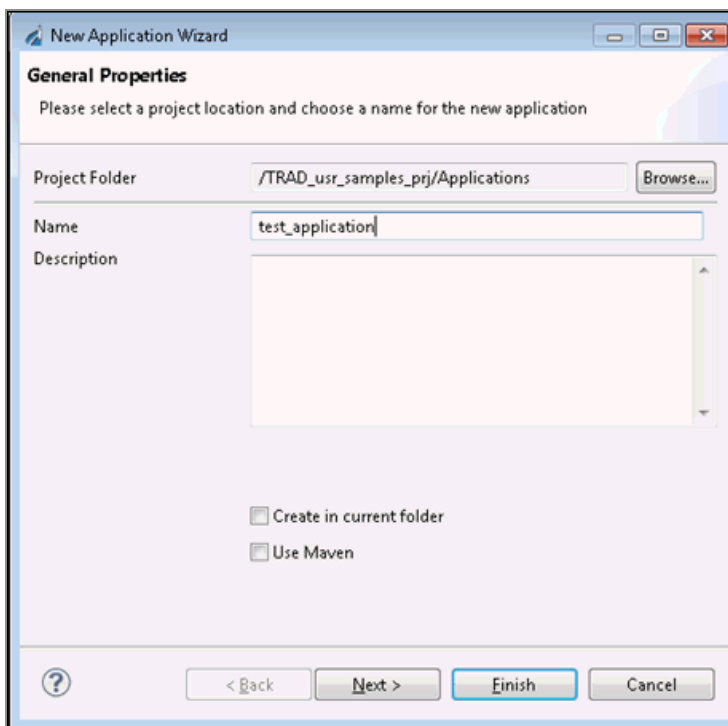
Rebuild Your Application

Procedure

1. In the Integration Explorer tab (left pane of iIT), right-click **Applications**, click **New**, and then select **Application** from the context menu, as shown in the following image.

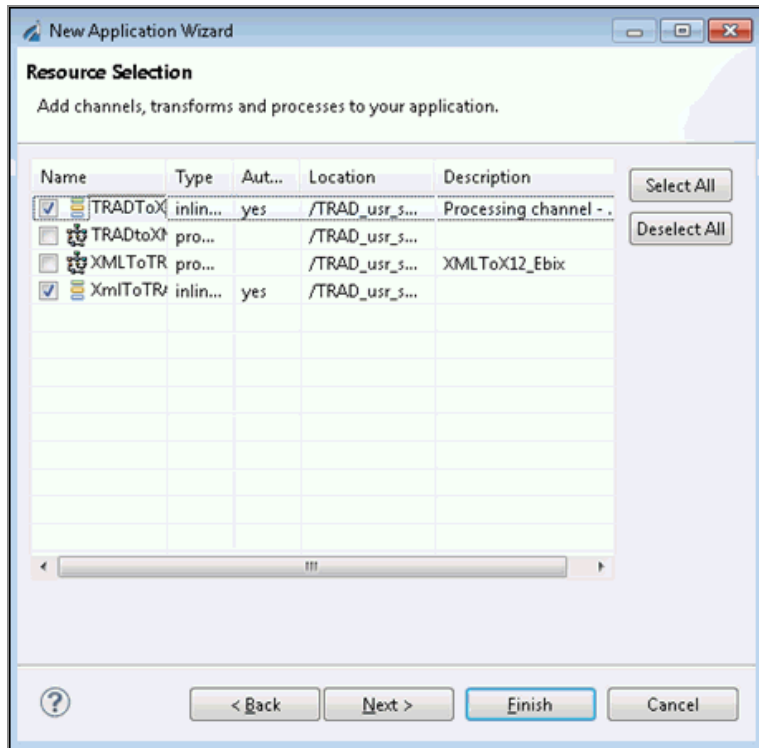


The New Application Wizard (General Properties pane) opens, as shown in the following image.



2. Enter a name for your application, and then click **Next**.

The Resource Selection pane opens, as shown in the following image.



3. Select the check boxes next to your channels and then click **Finish**.

! **Important:** Applications must be rebuilt and redeployed when changing Ebixes or SREGs in a channel. If you are unsure, rebuild and restart your application.

For more information on deploying and starting your application, see [Deployment Information for Your iWay Integration Solution](#).

Outbound Processing: XML to TRADACOMS

The iWay Integration Solution for TRADACOMS runs within iWay Service Manager (iSM). iSM validates an XML document based on TRADACOMS published implementation guides and converts it to a document in TRADACOMS format.

This chapter provides the information you need to understand and implement a basic outbound message flow.

Outbound Processing Overview

This section describes the iWay business components and the processing steps in the basic outbound message flow.

The outbound process converts an XML document to a TRADACOMS-formatted document. Outbound processing consists of the following components in the sample message flow:

- **Inlet**

The file **listener** picks up the incoming XML document.

- **Route and Process Flow**

A process flow guides the document through the next stages of the process.

Rules processing runs against the XML-formatted TRADACOMS document to validate its structure and content. The published TRADACOMS standards and user implementation guides define element types (for example, numeric, alpha, or date) and describe business rules to apply for validation.

The *XMLToTRADTransformAgent* obtains the message type and version from the XML-formatted TRADACOMS document. The appropriate transformation template is applied from the Ebix. The transformation converts the XML-formatted TRADACOMS document to TRADACOMS format.

The validation report service (*TRAD_Validation_rpt*) creates a validation report in XML format. This validation report indicates a status of *success* or *failure*, which is used to route the document in the process flow.

- **Outlet**

Outlets define how messages leave a channel at the end of a process. In this example, a *Passthrough* emitter is used. All the file emits are done in the process flow.

For more information on the emitters available with iSM, see the *iWay Service Manager User's Guide*.

For more information on deploying and starting the sample channels within the iWay Integration Application (iIA), see [Working With TRADACOMS Inbound and Outbound Applications Using iWay Integration Tools](#).

Special Register Sets

A Special Register (SREG) is a variable that users can set. As a best practice within the EDI framework, you can use SREGs to define input and output locations. When an iWay Integration Application (iIA) containing EDI channels is migrated between systems (for example, from *Dev* to *UAT*), the only changes required to deploy the iIA in the new system is to set the SREG and start the iIA. EDI channels use multiple locations and paths. This practice minimizes the migration effort. For a list of SREGs that are provided with iSM, see the *ibi™ iWay® Service Manager Programmer's Guide*. For more information on defining an SREG of your own, see the *ibi™ iWay® Service Manager User Guide*.

The sample outbound TRADACOMS channel sets all the SREGs that are listed in the following table.

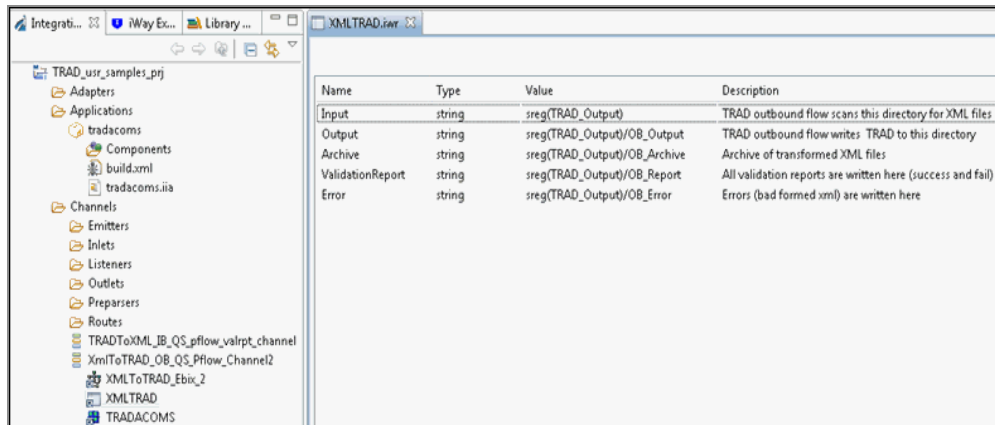
Name	Value
Input	sreg(TRAD_OUTPUT)
Output	sreg(TRAD_OUTPUT)
Archive	sreg(TRAD_OUTPUT)/OB_Archive
ValidationReport	sreg(TRAD_OUTPUT)/OB_Report
Error	sreg(TRAD_OUTPUT)/OB_Error

Modify SREG Sets in the Channel

Procedure

1. In the Integration Explorer tab (left pane of iIT), double-click the name of the SREG set (for example, XMLTRAD).

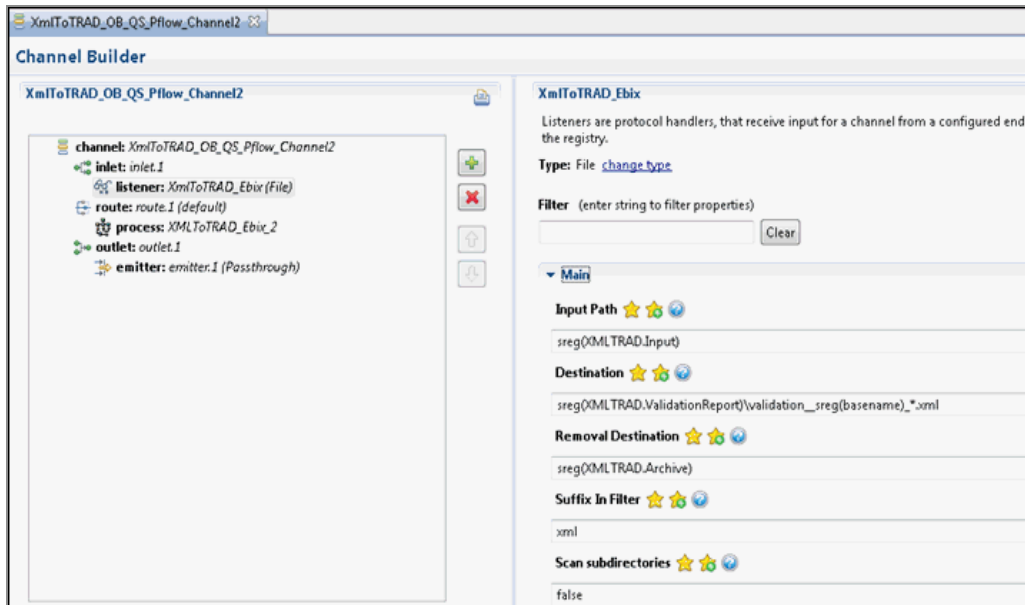
The selected SREG set opens as a new tab, as shown in the following image.



2. Ensure to save your changes.
3. If you make any changes to the SREG set, rebuild and redeploy the iIA for the changes to take effect.

Sample Channel File Listener

The following image shows a sample File listener that has been configured using the Channel Builder in iIT.



The following table describes the EDI-specific configuration parameters for the File listener. An asterisk (*) indicates that a parameter is required. For parameters not listed in the following table, accept the default value. Ensure that you have created any directories that are referenced by the SREGs, or errors occur during deployment.

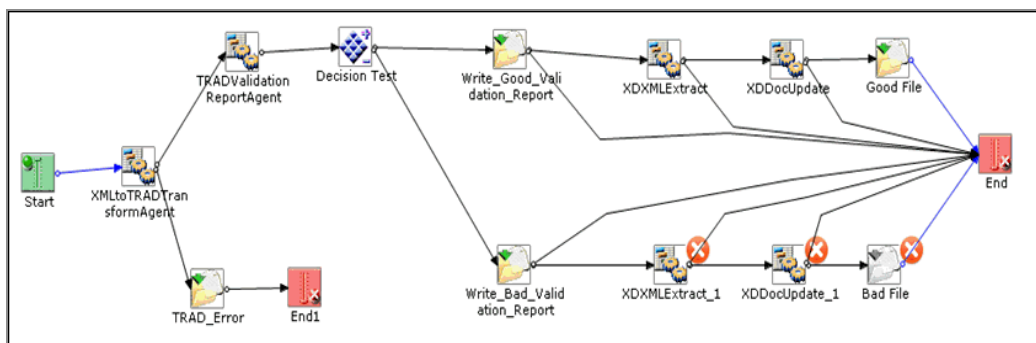
Parameter	Value
Input Path *	sreg(XMLTRAD.Input) This value is an SREG that uses a defined directory in which input messages are received.
Destination *	sreg(XMLTRAD.ValidationReport)\validation__sreg(backendname)_*.xml This value is an SREG that uses a defined directory in which output messages are received. Note: The underscore is a double underscore used in the destination to insert an underscore in the file name.
Removal Destination	sreg(XMLTRAD.Archive) This value is an SREG that uses a defined directory to

Parameter	Value
	<p>which output messages are moved if they fail during transformation.</p> <p>It is recommended to configure a removal destination when you are constructing a basic channel.</p>
Suffix In	<p>XML</p> <p>Input files with the extension .xml are allowed.</p>
Suffix Out	<p>TRAD</p> <p>In this example, the extension for output files is .trad.</p>

Sample Channel Process Flow

A route is defined in the sample channel that invokes the *XMLtoTRAD_Ebix_2* process flow. Using the validation report, valid transformed XML data is output to one folder and invalid transformed data will be written to another folder. If the transformation agent fails, then a *bad* validation report will be written. Bad validation reports can be routed through email or a queue for further inspection and remediation.

The following image shows the process flow from the sample channel.



Prior versions of the iWay e-Business user documentation demonstrate how to build channels and process flows in more detail using the iSM Administration Console and iIT.

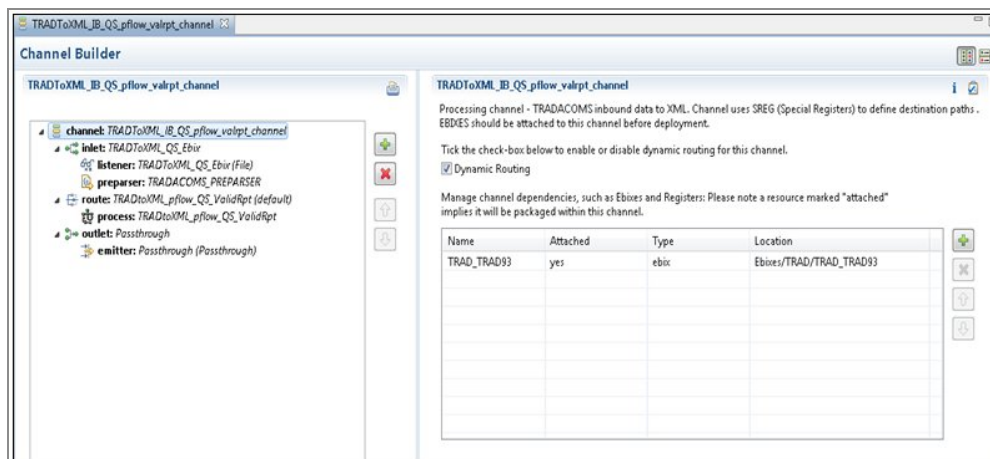
Adding an Ebix to a Channel

This section describes how to add an Ebix to the channel.

Add an Ebix to a Channel

Procedure

1. Open Channel Builder in iIT and then select the name of the channel in the upper-left pane.



2. Click the green plus (+) symbol to add a new Ebix. You can also select an Ebix and click the red (x) symbol to delete an Ebix.

If your channel contains multiple Ebixes, then use the up and down arrows to order the Ebixes. Ebixes are executed in the order that they appear on this screen, from top to bottom.

3. Ensure to save your changes.
4. If you make any changes to the channel, rebuild and redeploy the iIA for the changes to take effect.

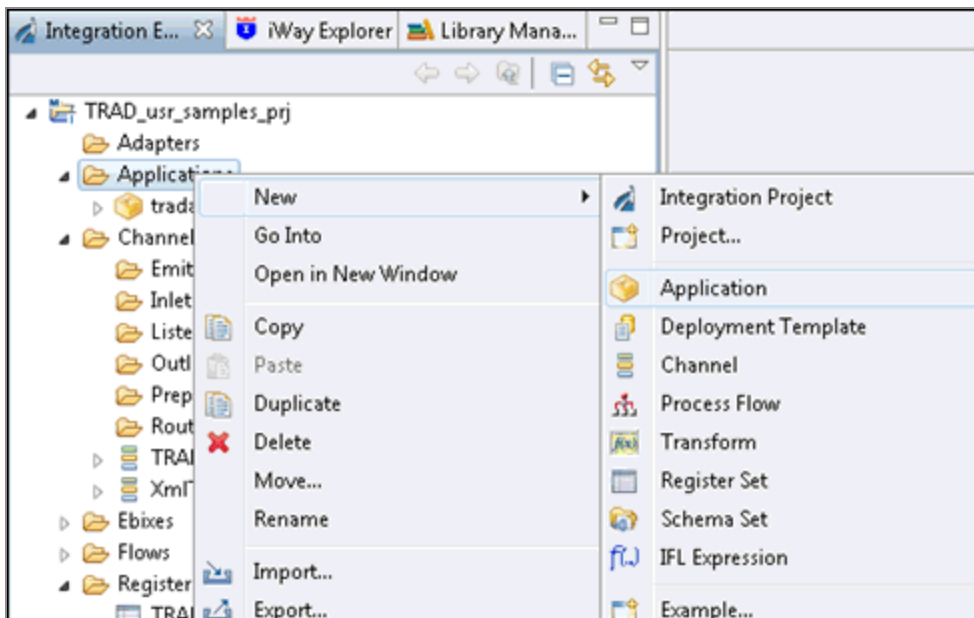
Rebuilding Your Application

If any changes are made to your application, then you must rebuild your application.

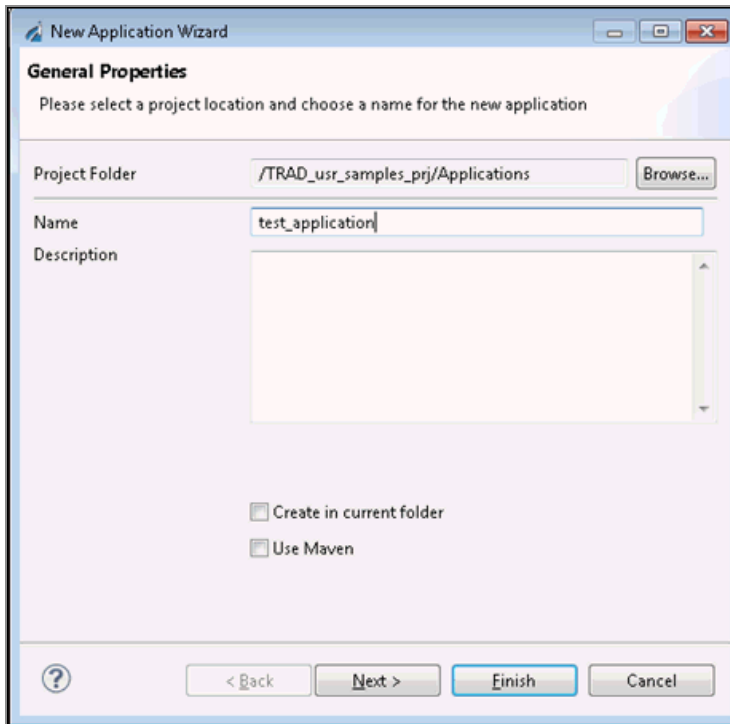
Rebuild Your Application

Procedure

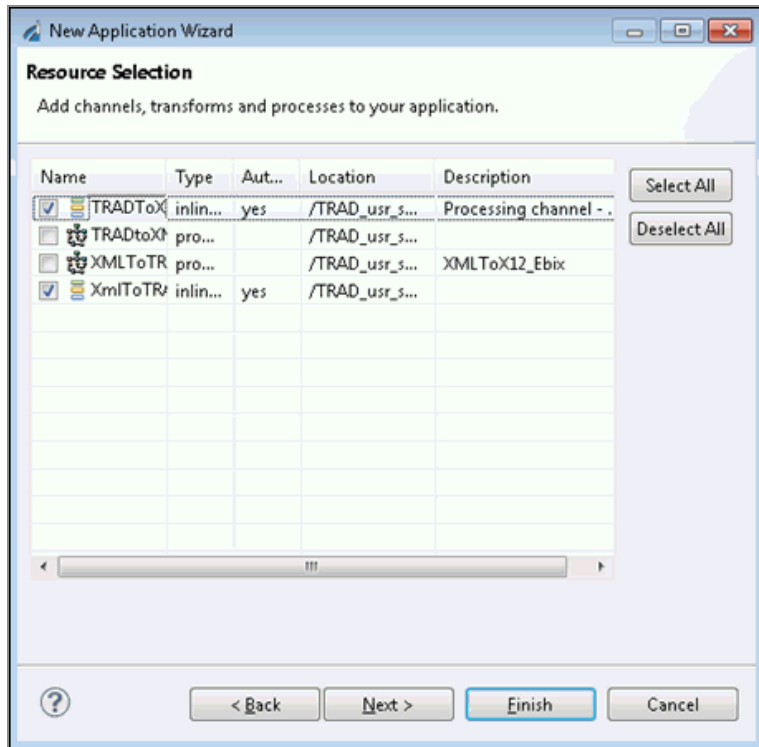
1. In the Integration Explorer tab (left pane of iIT), right-click **Applications**, click **New**, and then select **Application** from the context menu, as shown in the following image.



The New Application Wizard (General Properties pane) opens, as shown in the following image.



2. Enter a name for your application, and then click **Next**.
The Resource Selection pane opens, as shown in the following image.



3. Select the check boxes next to your channels and then click **Finish**.

! **Important:** Applications must be rebuilt and redeployed when changing Ebixes or SREGs in a channel. If you are unsure, rebuild and restart your application.

For more information on deploying and starting your application, see [Deployment Information for Your iWay Integration Solution](#).

Ebix-Supported Transaction Set

This appendix describes the TRADACOMS transaction set supported by the iWay Integration Solution for TRADACOMS in the Ebix file that is supplied with the product.

TRAD93

The iWay Integration Solution for TRADACOMS supports the **TRAD93** transaction set.

Configuring the EDI Activity Driver

This section describes how to configure the EDI Activity Driver using iWay Service Manager.

EDI Activity Driver Overview for TRADACOMS

The EDI Activity Driver is an extension of the Activity Facility in iWay Service Manager (iSM). It is used to log events as messages are processed. Logging can occur when:

- a message is acquired.
- a message is emitted.
- an error occurs.
- a component such as an agent or process flow is called.

For more information about the Activity Facility, see the *ibi™ iWay® Service Manager User Guide*.

Using the iSM Administration Console, you must first configure the EDI data provider and then the Activity Facility handler.

Configuring the EDI Data Provider

This section describes how to configure the EDI data provider using the iSM Administration Console.

Configure the EDI Data Provider

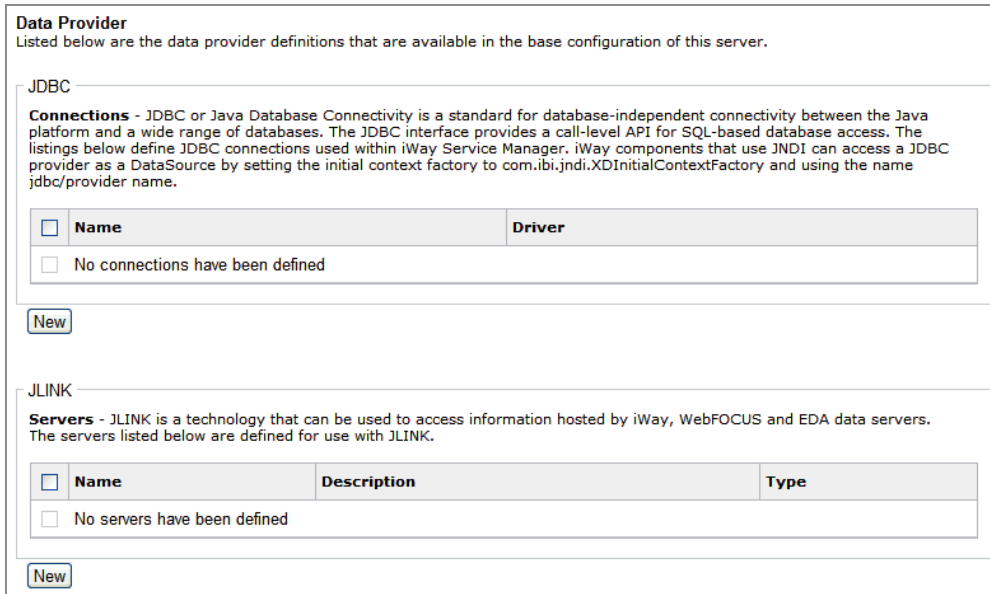
Procedure

1. Log in to the iSM Administration Console.

- From the left pane under Providers, click **Data Provider**, as shown in the following image.



The Data Provider pane opens, as shown in the following image.



The tables that are provided list the configured JDBC and JLINK data providers that are available. By default, no data providers are configured.

- In the JDBC area, click **New** to configure a new JDBC data provider.

The configuration pane for the JDBC Data Provider opens, as shown in the following image.

Data Provider - JDBC	
Listed below is the definition of the selected JDBC data provider. Add/Update the values as required.	
JDBC Connection Pool Properties	
Name *	Enter the name of the JDBC data provider to add. <input type="text" value="EDI_Activity_DB"/>
Driver Class	The JDBC driver class is the name of the class that contains the code for this JDBC Driver. <input type="text" value="com.mysql.jdbc.Driver"/> Select a predefined database or enter your own.
Connection URL	The JDBC connection URL to use when creating a connection to the target database. The URL generally includes the server name or IP address, the port or service, the data source name, and a driver specific prefix. <input type="text" value="jdbc:mysql://localhost:3306/IWay"/> Select a predefined connection URL template or enter your own.
User	User name with respect to the JDBC URL and driver. <input type="text" value="iway"/>
Password	Password with respect to the JDBC URL and driver. <input type="password" value="••••"/>
Connection Pool Properties	
Initial Pool Size *	Number of connections to place in the pool at startup. <input type="text" value="1"/>
Maximum Number of Idle Connections *	Maximum number of idle connections to retain in the pool. 0 means no limit except what is enforced by the maximum number of connections in the pool. <input type="text" value="1"/>
Maximum Number of Connections *	Maximum number of connections in the pool. 0 means no limit. <input type="text" value="1"/>
Login Timeout	Time in seconds to wait for a pooled connection before throwing an exception. 0 means wait forever. <input type="text"/>

- In the Name field, enter a name for the new JDBC data provider, for example, EDI_Activity_DB.
- From the Driver Class dropdown list, select an appropriate driver or enter the specific driver name (class) that you are using, for example:

```
com.mysql.jdbc.Driver
```

- From the Connection URL dropdown list, select an appropriate connection URL or enter the specific driver connection URL that you are using, for example:

```
jdbc:mysql://localhost:3306/IWay
```

- In the User field, enter a username with respect to the JDBC URL and driver.
- In the Password field, enter a password with respect to the JDBC URL and driver.
- In the Initial Pool Size field, enter the number of connections to place in the connection pool during startup.

- In the Maximum Number of Idle Connections field, enter the maximum number of idle connections to retain in the connection pool.

A value of zero means that there is no limit, except what is enforced by the maximum number of connections in the connection pool.

- In the Maximum Number of Connections field, enter the maximum number of connections in the connection pool.

A value of zero means that there is no limit.

- Click **Add**.

The JDBC data provider that you configured is added to the JDBC Connections list, as shown in the following image.

Data Provider
Listed below are the data provider definitions that are available in the base configuration of this server.

JDBC

Connections - JDBC or Java Database Connectivity is a standard for database-independent connectivity between the Java platform and a wide range of databases. The JDBC interface provides a call-level API for SQL-based database access. The listings below define JDBC connections used within iWay Service Manager. iWay components that use JNDI can access a JDBC provider as a DataSource by setting the initial context factory to com.ibi.jndi.XDInitialContextFactory and using the name jdbc/provider name.

<input type="checkbox"/>	Name	Driver
<input type="checkbox"/>	EDI_Activity_DB	com.mysql.jdbc.Driver

JLINK

Servers - JLINK is a technology that can be used to access information hosted by iWay, WebFOCUS and EDA data servers. The servers listed below are defined for use with JLINK.

<input type="checkbox"/>	Name	Description	Type
<input type="checkbox"/>	No servers have been defined		

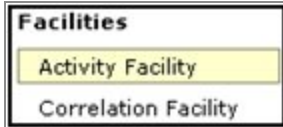
Configuring the EDI Activity Driver

This section describes how to configure the EDI Activity Driver using the iSM Administration Console.

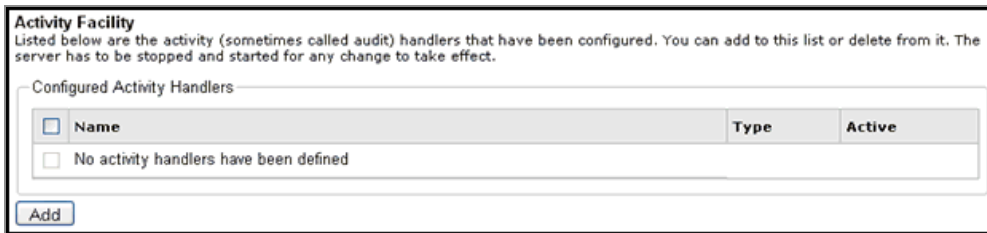
Configure the EDI Activity Driver

Procedure

1. Log in to the iSM Administration Console.
2. From the left pane under Facilities, click **Activity Facility**, as shown in the following image.



The Activity Facility pane opens, as shown in the following image.



The table that is provided lists the configured Activity Facility handlers. Initially, no handlers are shown.

3. Click **Add** to configure a new Activity Facility handler.

The configuration pane for the Activity Facility handler opens.

 A screenshot of the "Activity" configuration form. The form is titled "Activity" and contains several fields:

- Type:** A dropdown menu with the text "The type is the specific class of handler in use" above it. The selected value is "EDI Activity Logs".
- Name:** A text input field with the text "The handler will be known by this name in the system. Names must be unique." above it. The entered value is "EDI Activity Logger".
- Description:** A large text area with the text "Describe the purpose of this handler" above it. It is currently empty.
- Active:** A dropdown menu with the text "Active handlers perform work in the server. Inactive handlers remain defined but are not used during this server run. To change the active state, after updating you must cold restart the server." above it. The selected value is "true".

4. From the Type dropdown list, select **EDI Activity Logs**.
5. Enter a unique name for the EDI Activity Driver and a brief description.
6. From the Active dropdown list, select **true**.
7. Configure the JDBC driver for the database that you are using.

Configuration Parameters	
JNDI Factory Name	JNDI initial context factory class used to access data source. Use <code>com.ibi.jndi.XDInitialContextFactory</code> for an iWay JDBC provider or leave blank for JVM default. <input type="text" value="com.ibi.jndi.XDInitialContextFactory"/>
JNDI Name *	JNDI Name for the data source this driver will use. To use an iWay JDBC provider, enter the JNDI name as <code>jdbc/provider name</code> otherwise the defined provider's information will be used. <input type="text" value="jdbc/EDI_Activity_DB"/>
Table *	Table name to which to write log. <input type="text" value="IAM_ACTIVITY"/>
Compression	What form of compression, if any, should be used on the messages. Compression saves space at the expense of time. <input type="text" value="none"/> <input type="button" value="Pick one"/>

If the database tables do not exist, they are automatically created when the iSM is restarted.

8. Provide values for the remaining parameters, as defined in the following table.

Parameter Name	Type	Description
JNDI Factory Name	String	The JNDI initial context factory class that is used to access a data source. Use <code>com.ibi.jndi.XDInitialContextFactory</code> for an iWay JDBC provider or leave this field blank for the JVM default.
JNDI Name	String	The JNDI name for the data source this driver uses. To use an iWay JDBC provider, enter the JNDI name as <code>jdbc/<data provider name></code> , where <i>data provider name</i> is the name of the EDI Activity Driver that was specified in step 4. Otherwise the information for the defined provider is used.
Table	String	Table name for the activity log. This must be a valid identifier in the database being used. If the table does not exist at startup, it is created automatically.
Compression	Dropdown list	Specify whether the messages are to be

Parameter Name	Type	Description
		<p>compressed. Values include:</p> <ul style="list-style-type: none"> • none (default) • smallest • fastest • standard • Huffman
Start Events	Boolean Dropdown list	If set to true (default), the input messages are recorded in the activity log. This value must be set to true for use of the audit reports in the console.
Internal Events	Boolean Dropdown list	If set to true , system events are included in the activity log. System events include activities such as parsing and transformations (optional). False is selected by default.
Security Events	Boolean Dropdown list	If set to <i>true</i> (default), security events are recorded. This includes digital signature, and so on. However, console activity is not recorded.
Business Error Events	Boolean Dropdown list	If set to true , business errors are recorded, such as rules system violations. False is selected by default.
Emit Events	Boolean Dropdown list	If set to true (default), output messages from emitter services are recorded. This is required for use of the audit log reports in the console.

Parameter Name	Type	Description
End Events	Boolean Dropdown list	If set to true (default), the end of message processing is recorded in the activity log. This is required for use of the audit log reports in the console.
Notes Table	String	Table name for the notes table, which contains log annotations. If the table does not exist at startup, it is created automatically.
MAC Algorithm	String Dropdown list	The Message Authentication Code (MAC) algorithm. None (default) indicates that a MAC should not be computed.
MAC Provider	String Dropdown list	The Message Authentication Code (MAC) provider. Not Specified indicates that the default provider should be used. The remaining available value is SunJCE .
MAC Secret Key	String	The Message Authentication Code (MAC) secret key to use.

9. Click **Update**.

If necessary, start the database services.

10. Restart iSM to start the EDI Activity Driver and begin logging.

The EDI Activity Driver inserts records into the configured activity database. The records are designed for fast writing rather than for ease of later analysis. A set of inquiry service agents suitable for use in a process flow is available to assist during the analysis of the log. Users are cautioned that iWay does not guarantee the layout of the record from release to release, and this should be checked against the actual schema.

Database Field	Description
recordkey	Unique record identifier.
recordtype	Type of this record - the event being recorded. <ul style="list-style-type: none"> • 101 - Message start. • 131 - Entry to event (see subtype codes below). • 132 - Normal exit from event. • 133 - Failed exit from event. • 151 - Ancillary message (usually rules violation). • 181 - Emit. • 191 - Message end.
signature	Encoding of the listener name and protocol.
protocol	Name of the protocol.
address	Address to which an emit is to be issued. The format depends on the protocol.
tstamp	Timestamp of record.
correlid	Control Number
tid	Transaction ID assigned to this message.
msg	Message appropriate to this record type. For example, an input message contains the original message received, if possible. Streaming input does not contain a record.
context	Serialized special registers that were in the context at the time the record was written.

Database Field	Description
text	Message text for business errors (rules system violations).
status	Status code recorded. <ul style="list-style-type: none"> • 0 - Success • 1 - Success, message end (191 record) • 10 - Rules error
subtype	Event code for event records. <ul style="list-style-type: none"> • 1 - Preparser • 2 - Parser • 3 - In reviewer • 5 - In validation • 6 - In transform • 7 - Agent or flow • 8 - Out transform • 9 - Out validation • 11 - Premitter • 1000 - input record written to table before transformation
partner_to	Sender ID
partner_from	Receiver ID
encoding	Encoding of the listener that obtained the document.
mac	Not used in this version.
Driver version	1.0

Using iWay Integration Tools to Configure an Ebix for TRADACOMS

This section describes how to use iWay Integration Tools (iIT) to configure an e-Business Information Exchange (Ebix) file for TRADACOMS.

Ebix Overview

You can use iWay Integration Tools (iIT) to import, edit, export, and work with e-Business Information Exchange (Ebix) files. You can:

- Import and create a custom TRADACOMS Ebix in iIT.
- Add a custom qualifier to the custom Ebix.
- Export the edited Ebix to a physical location.

The edited Ebix can be returned and then tested with the appropriate TRADACOMS message.

Ebix Prerequisites

This section provides a list of prerequisites for using iWay Integration Tools (iIT) to configure an Ebix for TRADACOMS:

- Have a working knowledge of iIT and TRADACOMS.
- Ensure the iWay TRADACOMS adapter is installed.
- Ensure iIT Version 8.0 is installed.

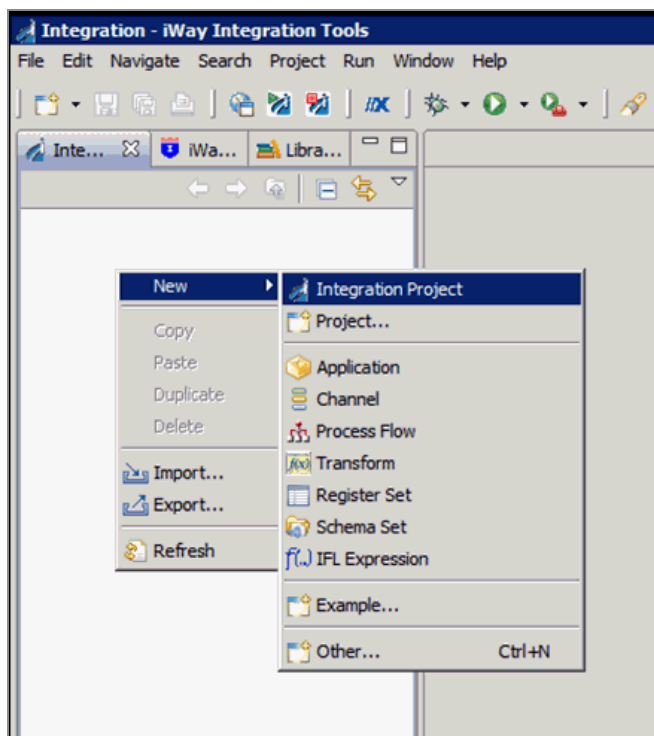
Working With iWay Integration Tools (iIT)

This section describes how to import, edit, and export an Ebix using iWay Integration Tools (iIT).

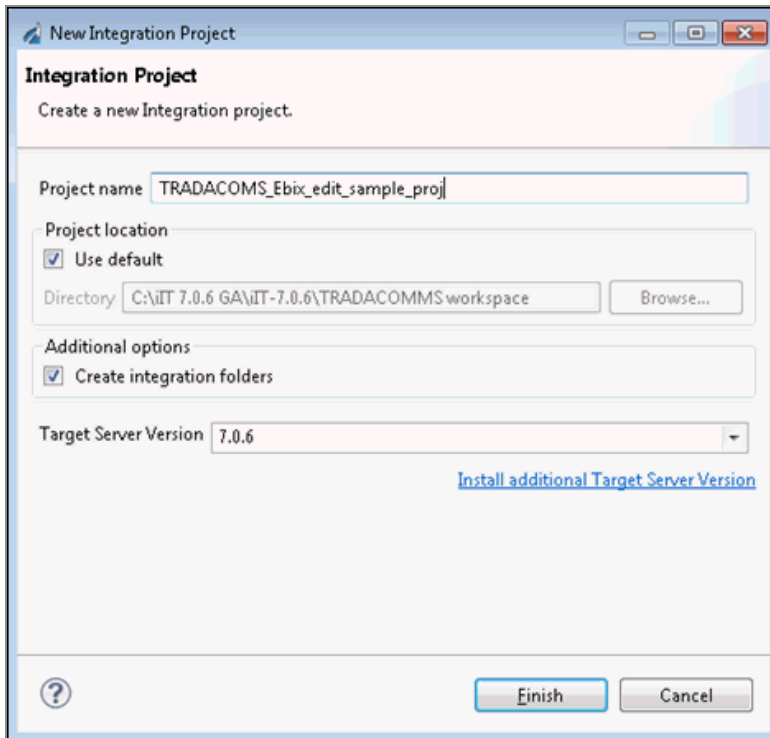
Import an Ebix

Procedure

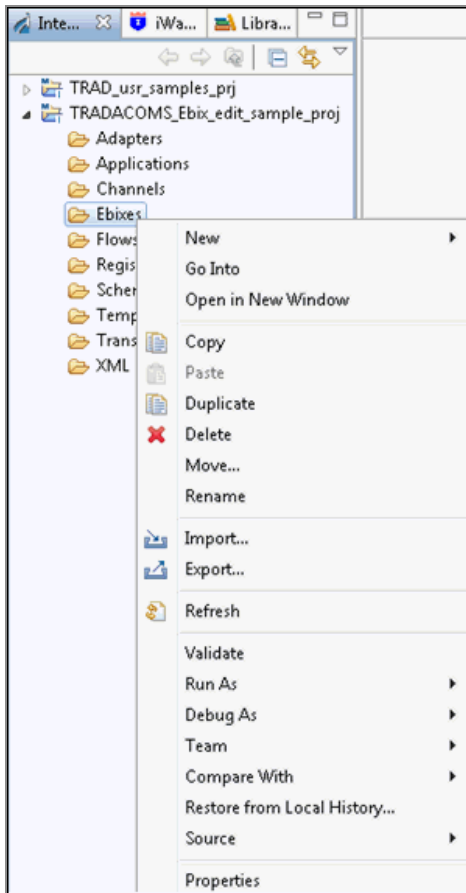
1. Start iWay Integration Tools (iIT).
2. Right-click the Integration Explorer pane, click **New**, and then select **Integration Project** from the context menu, as shown in the following image.



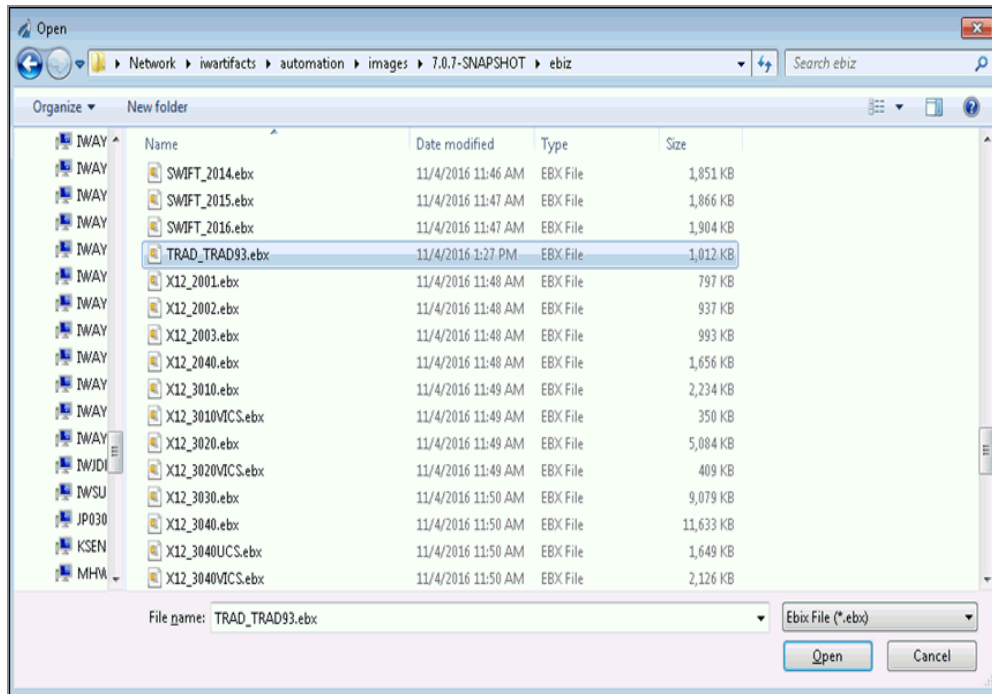
3. Enter a new Integration Project name, for example, *TRADACOMS_Ebix_edit_sample_proj*, in the Project name field, and then click **Finish**, as shown in the following image.



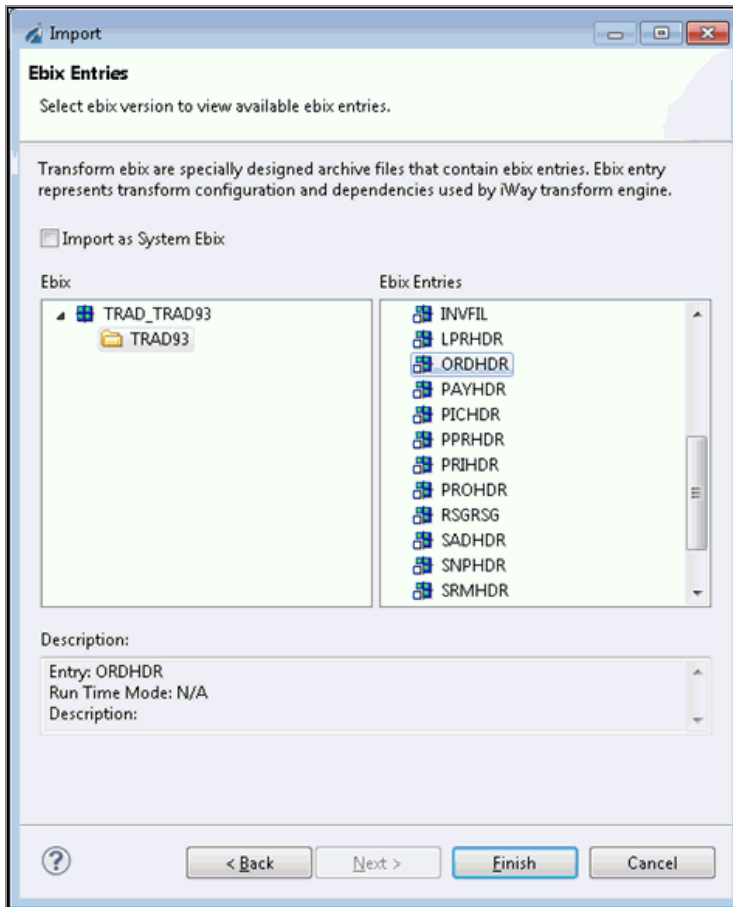
4. Right-click the Integration Explorer pane and select **Import** from the context menu, as shown in the following image.



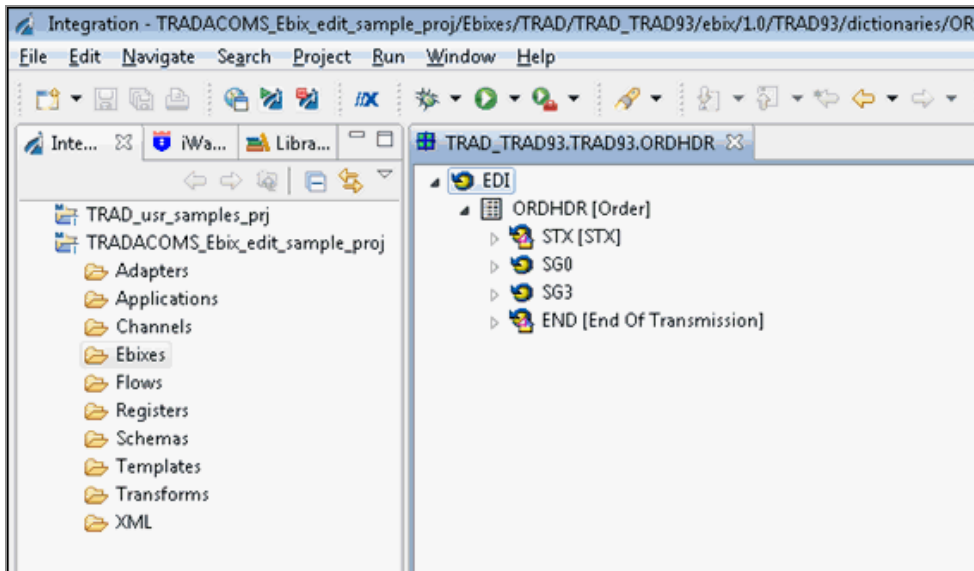
5. In the Import wizard, expand **iWay Integration**, select **Ebix**, and then click **Next**.
6. Click the **ellipsis (...)** button to the right of the Import field.
The Open dialog is displayed.
7. Navigate to and select your TRAD_TRAD93.ebx file, and then click **Open**.



8. Click **Next**.
9. In the Ebix pane, expand the Ebix tree, and then in the Ebix Entries pane, highlight **ORDHDR**, and then click **Finish**.



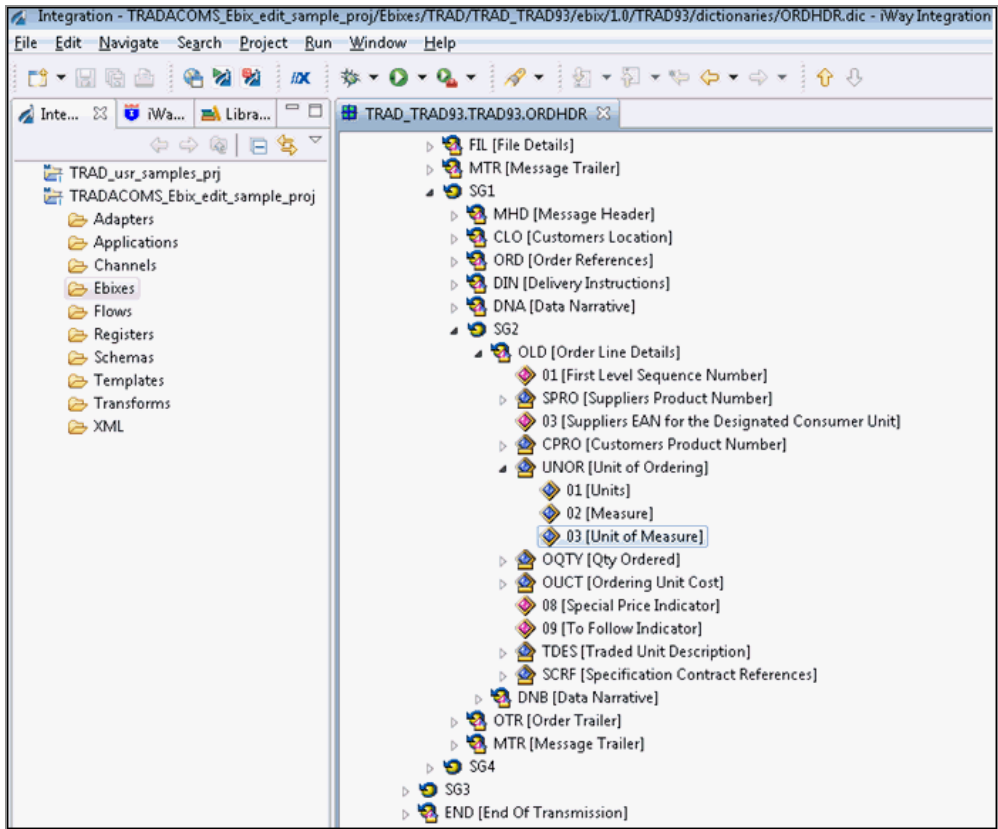
Your iIT interface should now resemble the following image:



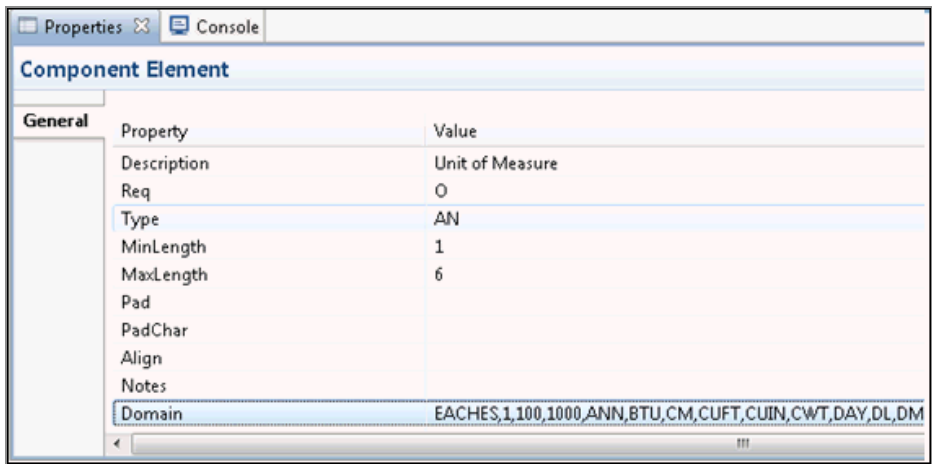
Edit an Ebix

Procedure

1. Expand **SG0**, **SG1**, **SG2**, **OLD**, **UNOR**, and then select **03 [Unit of Measure]**, as shown in the following image.

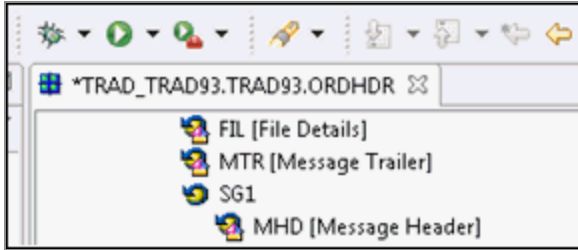


2. Click the Properties tab at the bottom, then scroll down to view the Domain value, and add **EACHES** into the Domain value field in the properties window.

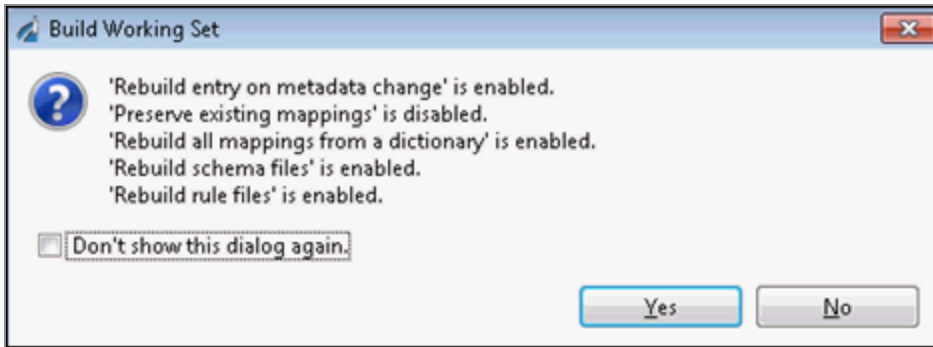


3. Save your edited Ebix by clicking the **Save** icon.
An asterisk (*) character appears next to the file name until you have saved the

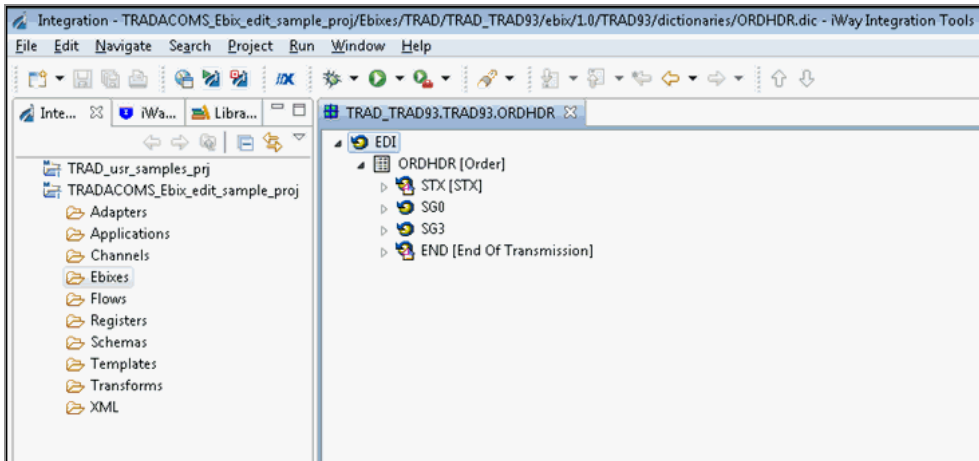
edited changes, as shown in the following image.



4. Click **Yes** to confirm your changes.



Your iIT interface should now resemble the following image:



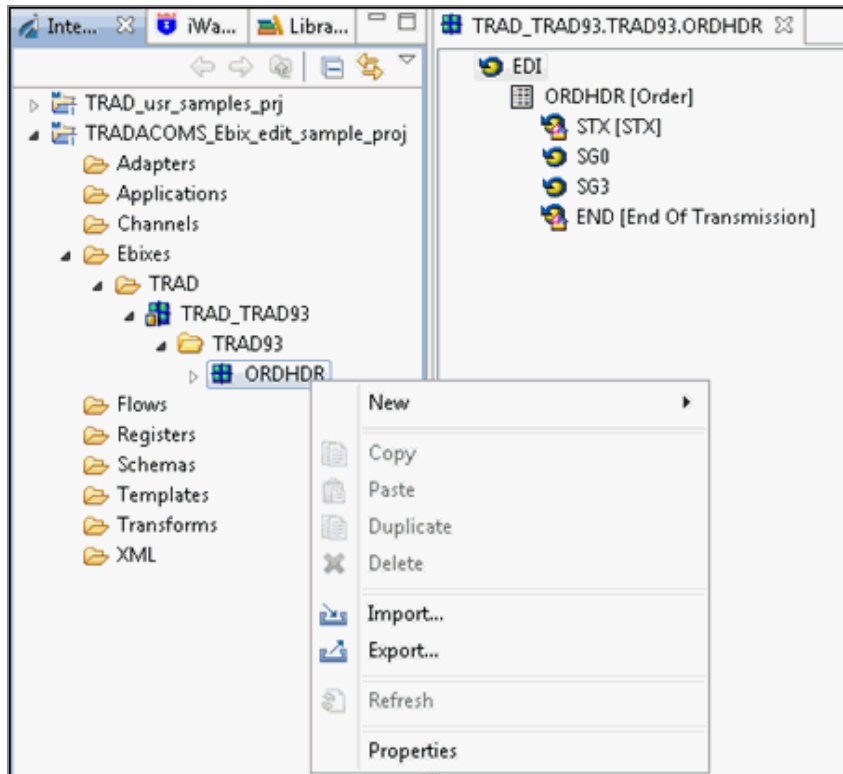
Note: The asterisk (*) character disappears once the edited Ebix has been saved successfully.

Export an Ebix

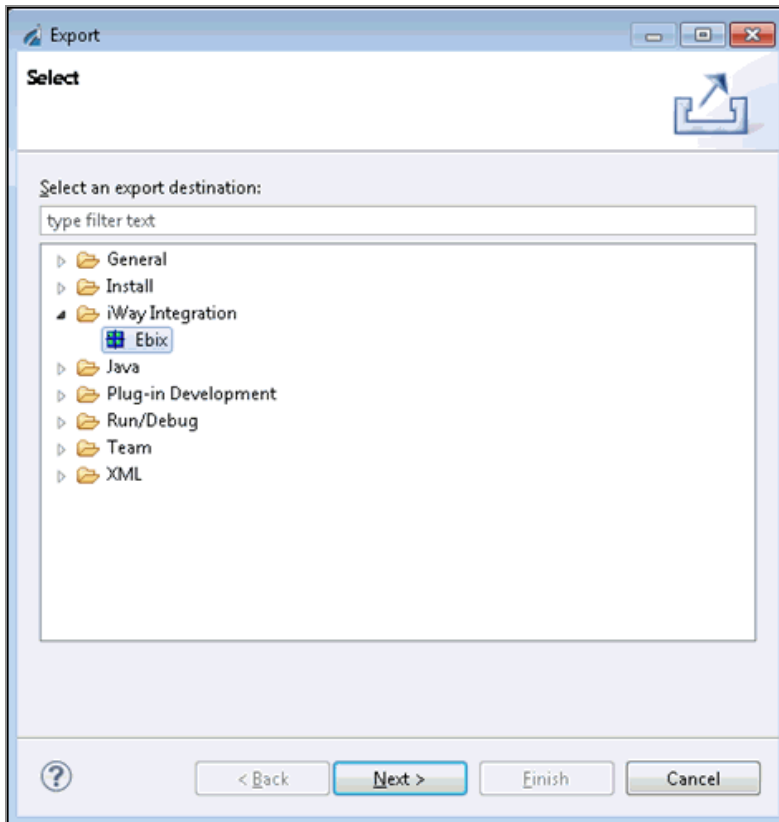
To export an Ebix:

Procedure

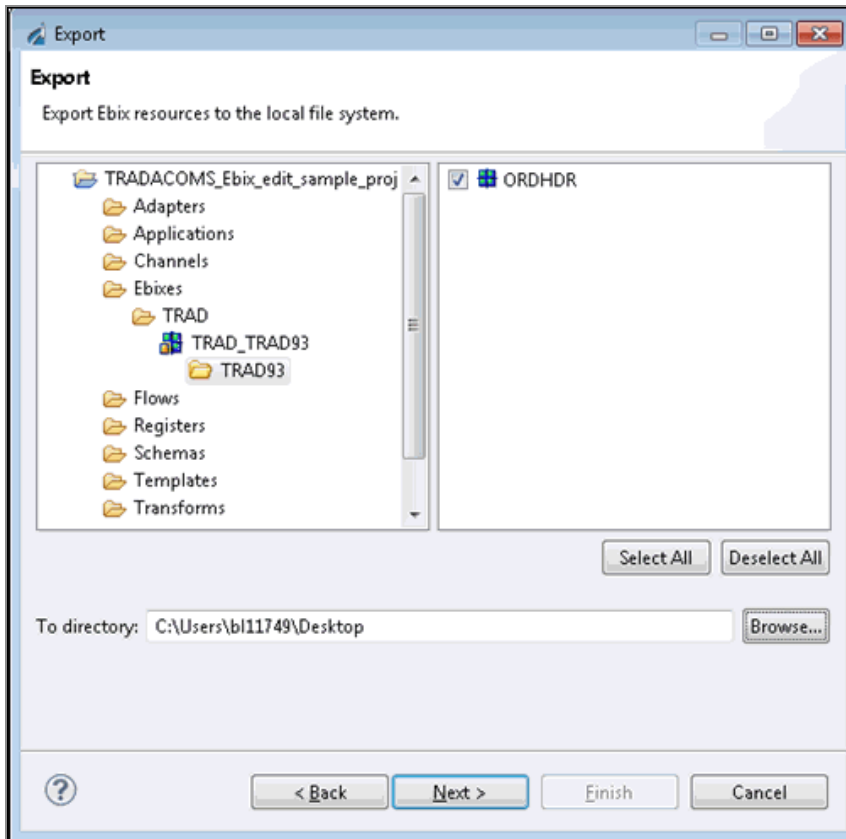
1. Expand the **Ebixes** folder in the left pane, then expand **TRAD**, **TRAD_TRAD93**, **Trad93**, and then right-click **ORDHDR** and select **Export** from the context menu.



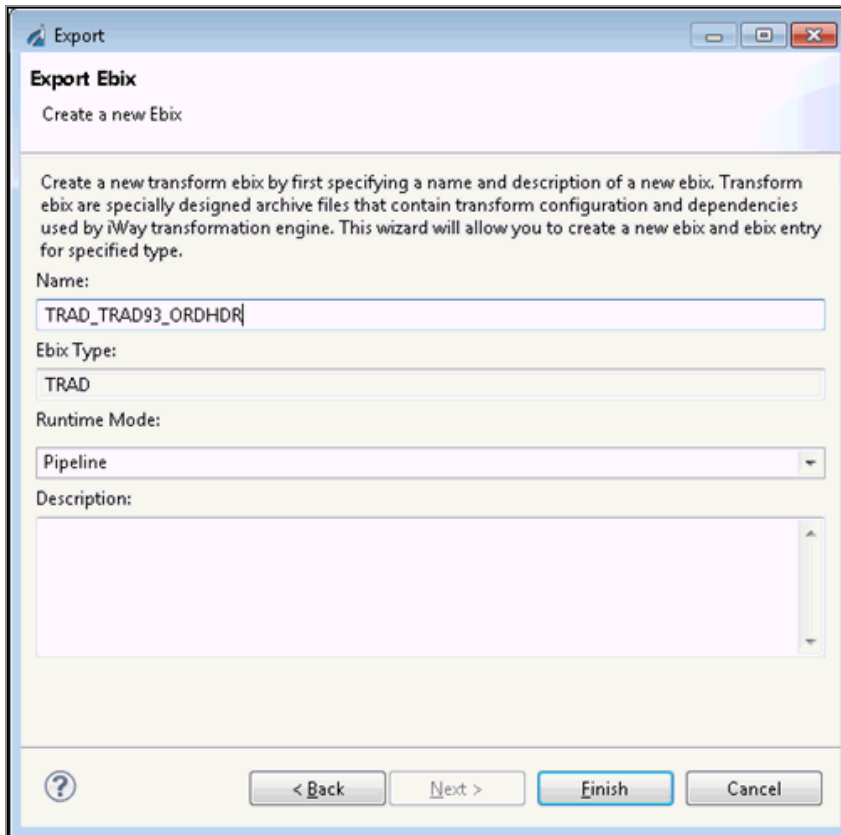
2. Expand the **iWay Integration** folder, select **Ebix**, and then click **Next**, as shown in the following image.



3. In the left pane, expand **TRADACOMS_Ebix_edit_sample_proj**. Then on the right pane, select the checkbox for the **ORDHDR** document and enter or select the destination to write the Ebix, as shown in the following image.



4. Click **Next**.
5. Provide a name for the Ebix in the Name field, add an optional description (optional), and then click **Finish**.



The screenshot shows a dialog box titled "Export Ebix" with the subtitle "Create a new Ebix". The main text explains that transform ebix are specially designed archive files containing transform configuration and dependencies used by the iWay transformation engine. The wizard allows creating a new ebix and ebix entry for a specified type.

Fields and controls:

- Name:** A text input field containing "TRAD_TRAD93_ORDHDR".
- Ebix Type:** A dropdown menu with "TRAD" selected.
- Runtime Mode:** A dropdown menu with "Pipeline" selected.
- Description:** A large empty text area.
- Buttons:** "< Back", "Next >", "Finish", and "Cancel".
- Help:** A question mark icon in a circle.

Your exported Ebix is now available in the specified location.

You can attach the Ebix to your channel in iIT using the Channel Builder. Be sure to set the execution order of your Ebix after adding it to your queue. The order that the Ebix are executed is top-down as they are displayed. If you want this custom ORDHDR dictionary to be executed prior to the standard one in the stock Ebix, then the custom Ebix must be at the top of the list.

Sample TRADACOMS Files

This appendix provides sample TRADACOMS files.

TRADACOMS ORDHDR (Order Header)

TRADACOMS uses a header with each interchange (ORDHDR for the ORDERS message). This header message contains information that is specific to each message in the interchange.

The following is a sample TRADACOMS ORDHDR (Order Header).

```
STX=ANA:1+50000000000000:SOME STORES LTD+50100000000000:SUPPLIER UK
LTD+070315:130233+000007+PASSW+ORDHDR+B'
MHD=1+ORDHDR:9' TYP=0430+NEW-ORDERS'
SDT=50100000000000:000030034' CDT=50000000000000' FIL=1630+1+070315'
MTR=6' MHD=2+ORDERS:9'
CLO=50000000000283:89828+EAST SOMEWHERE DEPOT' ORD=70970::070315'
DIN=070321++0000' OLD=1+50102100000000++:00893592+12+60++++CRUSTY ROLLS:4
PACK' OTR=1'
MTR=7' MHD=3+ORDTLR:9' OFT=1'
MTR=3' END=3'
```

TRADACOMS INVFIL (Invoice File Header)

The following is a sample TRADACOMS INVFIL (Invoice File Header).

```
STX=ANAA:1+501xxxxxxxxxx:name+501xxxxxxxxxx:name+040316:184411+00001++IN
VFIL'
MHD=1+INVFIL:9'
TYP=0700+INVOICES'
SDT=1234567895011:053752CF01STDD+name+address:Co. Antrim:N.Ireland:OG27
5WQ+412557175'
CDT=1234567895011+name+address:Ehorel:Birmingham:OG45 9CM'
FIL=1+1+160901'
FDT=040302+040302'
MTR=7'
```

```
MHD=2+INVOIC:9'  
CLO=0000000100007:123++GOLF CLUB:ABEGU ROAD:PNEEV PX SRETHF::OG388YC'  
IRF=01000589M+040302+040302'  
ODD=1+::040302+:040302'  
ILD=1+1+:8408++:31266+0+2:2500:KG+54300+135800+Z+0+++SILVERSIDE 1-5 KG'  
STL=1+Z+0+0+1358+++++1358++0+0++1358'  
TLR=1+1358+++++1358++1358+0++1358'  
MTR=8'  
MHD=3+VATTLR:9'  
VRS=1+Z+0+1358+1358+0+1358+1358'  
MTR=3'  
MHD=4+INVTLR:9'  
TOT=1358+1358+0+1358+1358+1'  
MTR=3'  
MHD=5+RSGRSG:2'  
RSG=00001+501xxxxxxxxxxx'  
MTR=3'  
END=5'
```

ibi Documentation and Support Services

For information about this product, you can read the documentation, contact Support, and join Community.

How to Access ibi Documentation

Documentation for ibi products is available on the [Product Documentation website](#), mainly in HTML and PDF formats.

The [Product Documentation website](#) is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

The documentation for this product is available on the [ibi™ iWay® Service Manager Documentation](#) page.

How to Contact Support for ibi Products

You can contact the Support team in the following ways:

- To access the Support Knowledge Base and getting personalized content about products you are interested in, visit our [product Support website](#).
- To create a Support case, you must have a valid maintenance or support contract with a Cloud Software Group entity. You also need a username and password to log in to the [product Support website](#). If you do not have a username, you can request one by clicking **Register** on the website.

How to Join ibi Community

ibi Community is the official channel for ibi customers, partners, and employee subject matter experts to share and access their collective experience. ibi Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from ibi products. For a free registration, go to [ibi Community](#).

Legal and Third-Party Notices

SOME CLOUD SOFTWARE GROUP, INC. (“CLOUD SG”) SOFTWARE AND CLOUD SERVICES EMBED, BUNDLE, OR OTHERWISE INCLUDE OTHER SOFTWARE, INCLUDING OTHER CLOUD SG SOFTWARE (COLLECTIVELY, “INCLUDED SOFTWARE”). USE OF INCLUDED SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED CLOUD SG SOFTWARE AND/OR CLOUD SERVICES. THE INCLUDED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER CLOUD SG SOFTWARE AND/OR CLOUD SERVICES OR FOR ANY OTHER PURPOSE.

USE OF CLOUD SG SOFTWARE AND CLOUD SERVICES IS SUBJECT TO THE TERMS AND CONDITIONS OF AN AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER AGREEMENT WHICH IS DISPLAYED WHEN ACCESSING, DOWNLOADING, OR INSTALLING THE SOFTWARE OR CLOUD SERVICES (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH LICENSE AGREEMENT OR CLICKWRAP END USER AGREEMENT, THE LICENSE(S) LOCATED IN THE “LICENSE” FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE SAME TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of Cloud Software Group, Inc.

ibi, the ibi logo, iWay, and TIBCO are either registered trademarks or trademarks of Cloud Software Group, Inc. in the United States and/or other countries.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only. You acknowledge that all rights to these third party marks are the exclusive property of their respective owners. Please refer to Cloud SG’s Third Party Trademark Notices (<https://www.cloud.com/legal>) for more information.

This document includes fonts that are licensed under the SIL Open Font License, Version 1.1, which is available at: <https://scripts.sil.org/OFL>

Copyright (c) Paul D. Hunt, with Reserved Font Name Source Sans Pro and Source Code Pro.

Cloud SG software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the “readme” file for the availability of a specific version of Cloud SG software 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. CLOUD SG MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S), THE PROGRAM(S), AND/OR THE SERVICES DESCRIBED IN THIS DOCUMENT AT ANY TIME WITHOUT NOTICE.

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 "README" FILES.

This and other products of Cloud SG may be covered by registered patents. For details, please refer to the Virtual Patent Marking document located at <https://www.cloud.com/legal>.

Copyright © 2021-2024. Cloud Software Group, Inc. All Rights Reserved.