TIBCO ActiveMatrix® Adapter Service Engine for IBM i

Examples

Software Release 6.0
March 2010

TIBCO®
The Power of Now®

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 document contains confidential information that 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 TIBCO Software Inc.

TIB, TIBCO, TIBCO Adapter, TIBCO ActiveMatrix, Predictive Business, Information Bus, The Power of Now, TIBCO ActiveMatrix BusinessWorks, TIBCO Rendezvous, TIBCO Administrator, TIBCO Designer, TIBCO Runtime Agent, TIBCO Hawk, TIBCO Enterprise Message Service, TIBCO Designer Add-in for TIBCO Business Studio, TIBCO ActiveMatrix Service Grid, TIBCO ActiveMatrix Service Bus, TIBCO ActiveMatrix BusinessWorks Service Engine, TIBCO ActiveEnterprise, and TIBCO Business Studio, are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

EJB, Java EE, J2EE, and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

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

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

Copyright © 2008-2010 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information

Contents

Figures v

Preface vii

Related Documentation viii
TIBCO ActiveMatrix Adapter Service Engine for IBM i Documentation viii
Other TIBCO Product Documentation ix
Third-Party Documentation. ix

Typographical Conventions x

How to Contact TIBCO Support. xiii

Chapter 1 Introduction. 1

Overview 2

Prerequisites 3
Loading a Library on an IBM System i Machine 3
Setting Up the Environment for TIBCO ActiveMatrix 3

Chapter 2 Publisher and RPC Client Example 7

Example Description 8

Setting Up the Example 9

Running the Example 15

Expected Results. 18

Chapter 3 Subscriber (Program Call) and RPC Server (Data Queue Write) Example 21

Example Description 22

Setting Up the Example 23

Running the Example 29

Expected Results. 32

Index 33

Figures

Figure 1 Receive CustomerMessage Process 13
Figure 2 Receive Customer Information Request from IBM System i 13
Figure 3 Published Message 18
Figure 4 Request Message 19
Figure 5 Call Program to Update Customer 25
Figure 6 Invoke Partner 26
Figure 7 Send HTTP Response 26
Figure 8 Invoke Server to Inquire Customer 27
Figure 9 Service Units of the Inbound Sample 30

Preface

TIBCO ActiveMatrix Adapter Service Engine for IBM i allows one-way (publish or subscribe) or two-way (request-response) message exchange between TIBCO environment and applications running on IBM System i machines.

This manual presents examples that demonstrate key adapter features. Work through these examples to get a hands-on understanding of how the adapter works.

Topics

- Related Documentation, page viii
- Typographical Conventions, page x
- How to Contact TIBCO Support, page xiii

Related Documentation

This section lists documentation resources you may find useful.

TIBCO ActiveMatrix Adapter Service Engine for IBM i Documentation

The following documents form the TIBCO ActiveMatrix Adapter Service Engine for IBM i documentation set:

- *TIBCO ActiveMatrix Adapter Service Engine for IBM i Installation* Read this manual for instructions on site preparation and installation.
- *TIBCO ActiveMatrix Adapter Service Engine for IBM i Configuration and Deployment* Read this manual for instructions on creating and configuring adapter projects. Instructions on how to deploy the projects are also included.
- *TIBCO ActiveMatrix Adapter Service Engine for IBM i Examples* Read this manual to work through the examples provided with the adapter service engine.
- *TIBCO ActiveMatrix Adapter Service Engine for IBM i Release Notes* Read the release notes for a list of new and changed features. This document also contains lists of closed and known issues for this release.

Before TIBCO ActiveMatrix Adapter Service Engine for IBM i can be installed, you must first install the TIBCO ActiveMatrix Adapter for IBM i. The following documents form the standalone adapter documentation set:

- *TIBCO ActiveMatrix Adapter for IBM i Concepts* Read this manual before reading any other book in the documentation set to familiarize yourself with the product and its uses.
- *TIBCO ActiveMatrix Adapter for IBM i Installation* Read this manual for instructions on site preparation and installation.
- *TIBCO ActiveMatrix Adapter for IBM i Configuration and Deployment* Read this manual for instructions on creating and configuring standalone adapter projects. Information on deploying adapter projects is also included.
- *TIBCO ActiveMatrix Adapter for IBM i Examples* Read this manual to work through the examples provided with the adapter.
- *TIBCO ActiveMatrix Adapter for IBM i Release Notes* Read the release notes for a list of new and changed features. This document also contains lists of closed and known issues for this release.

Other TIBCO Product Documentation

You may find it useful to read the documentation for the following TIBCO products:

- TIBCO ActiveMatrix® Administrator
- TIBCO ActiveMatrix BusinessWorks™
- TIBCO ActiveMatrix BusinessWorks™ Service Engine
- TIBCO ActiveMatrix® Service Grid
- TIBCO ActiveMatrix® Service Bus
- TIBCO Business Studio™
- TIBCO Enterprise Message Service™
- TIBCO Hawk®
- TIBCO Rendezvous®
- TIBCO Runtime Agent™

Third-Party Documentation

You may also find it useful to read the IBM documentation on the following web sites:

- http://www-03.ibm.com/systems/i/software/toolbox/
- http://jt400.sourceforge.net/

Typographical Conventions

The following typographical conventions are used in this manual.

Table 1 General Typographical Conventions

Convention	Use
TIBCO_HOME *ENV_HOME* *ADAS400_HOME*	Many TIBCO products must be installed within the same home directory. This directory is referenced in documentation as *TIBCO_HOME*. The value of *TIBCO_HOME* depends on the operating system. For example, on Windows systems, the default value is `C:\tibco`.
AMX_HOME	Other TIBCO products are installed into an installation environment. Incompatible products and multiple instances of the same product are installed into different installation environments. The directory into which such products are installed is referenced in documentation as *ENV_HOME*. The value of *ENV_HOME* depends on the operating system. For example, on Windows systems the default value is `C:\tibco`. TIBCO ActiveMatrix Adapter for IBM i installs into a directory within *TIBCO_HOME*. This directory is referenced in documentation as *ADAS400_HOME*. The value of *ADAS400_HOME* depends on the operating system. For example on Windows systems, the default value is `C:\tibco\adapter\adas400\6.0`. TIBCO ActiveMatrix installs into a directory within *TIBCO_HOME*. This directory is referenced in documentation as *AMX_HOME*. The value of *AMX_HOME* depends on the operating system. For example on Windows systems, the default value is `C:\tibco\amx\`. TIBCO ActiveMatrix Adapter Service Engine for IBM i installs in a directory within *AMX_HOME*.
`code font`	Code font identifies commands, code examples, filenames, pathnames, and output displayed in a command window. For example: Use `MyCommand` to start the foo process.
`bold code font`	Bold code font is used in the following ways: • In procedures, to indicate what a user types. For example: Type **`admin`**. • In large code samples, to indicate the parts of the sample that are of particular interest. • In command syntax, to indicate the default parameter for a command. For example, if no parameter is specified, `MyCommand` is enabled: `MyCommand [`**`enable`**` | disable]`

Table 1 General Typographical Conventions (Cont'd)

Convention	Use
italic font	Italic font is used in the following ways: • To indicate a document title. For example: See *TIBCO ActiveMatrix BusinessWorks Concepts*. • To introduce new terms For example: A portal page may contain several portlets. *Portlets* are mini-applications that run in a portal. • To indicate a variable in a command or code syntax that you must replace. For example: `MyCommand` *PathName*
Key combinations	Key name separated by a plus sign indicate keys pressed simultaneously. For example: Ctrl+C. Key names separated by a comma and space indicate keys pressed one after the other. For example: Esc, Ctrl+Q.
	The note icon indicates information that is of special interest or importance, for example, an additional action required only in certain circumstances.
	The tip icon indicates an idea that could be useful, for example, a way to apply the information provided in the current section to achieve a specific result.
	The warning icon indicates the potential for a damaging situation, for example, data loss or corruption if certain steps are taken or not taken.

Table 2 Syntax Typographical Conventions

Convention	Use
`[ ]`	An optional item in a command or code syntax. For example: `MyCommand [optional_parameter] required_parameter`
`|`	A logical `OR` that separates multiple items of which only one may be chosen. For example, you can select only one of the following parameters: `MyCommand para1 | param2 | param3`

Table 2 Syntax Typographical Conventions (Cont'd)

Convention	Use

`{ }`	A logical group of items in a command. Other syntax notations may appear within each logical group.
	
	For example, the following command requires two parameters, which can be either the pair `param1` and `param2`, or the pair `param3` and `param4`.
	
	`MyCommand {param1 param2} | {param3 param4}`
	
	In the next example, the command requires two parameters. The first parameter can be either `param1` or `param2` and the second can be either `param3` or `param4`:
	
	`MyCommand {param1 | param2} {param3 | param4}`
	
	In the next example, the command can accept either two or three parameters. The first parameter must be `param1`. You can optionally include `param2` as the second parameter. And the last parameter is either `param3` or `param4`.
	
	`MyCommand param1 [param2] {param3 | param4}`

How to Contact TIBCO Support

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

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

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

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

  https://support.tibco.com

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

Chapter 1 Introduction

This chapter contains an overview of and the prerequisites for the examples. Details about how to use the examples are explained in later chapters of this book.

Topics

- Overview, page 2
- Prerequisites, page 3

Overview

The examples shipped with the installation package demonstrate how to configure the adapter with Publication Service, Request-Response Invocation Service, Request-Response Service, and Subscription Services.

- Though all of the examples demonstrate adapter functionality for customer details, they are not related. Each example should be considered separately.
- Do not change the names given in the examples when trying the example scenarios. Changing names may result in example scenarios not working correctly.

Details for the examples are as follows:

- Chapter 2, Publisher and RPC Client Example demonstrates a Publisher that publishes a message from an application program running on the IBM System i machine; and a RPC Client that sends a request from an application running on an IBM System i machine and receives a reply with the requested information.
- Chapter 3, Subscriber (Program Call) and RPC Server (Data Queue Write) Example demonstrates a Subscriber that calls an application program on an IBM System i machine to update a customer record; and explains how to configure the adapter to write a message to a data queue on an IBM System i machine in synchronous mode.

Prerequisites

Before running the examples, you need to load a library on an IBM System i machine, and to set up the environment for TIBCO ActiveMatrix.

To run the examples, which are wired with BWSE, you must install TIBCO Designer Add-in for TIBCO business Studio and TIBCO ActiveMatrix BusinessWorks Service Engine.

Loading a Library on an IBM System i Machine

For the adapter to interact with an IBM System i machine, you need to load a saved file that contains pre-defined libraries to the IBM System i machine.

1. On the machine where the adapter is installed, start a command session. Change the directory of the command session to *ADAS400_HOME*`\examples`.
2. Start an FTP session in the command session to the IBM System i machine.
3. In the FTP session, change the transfer mode to binary using the command **`bin`**.
4. Transfer the saved file named `ADAS400.SAVF` from the *ADAS400_HOME*`\examples` directory to the IBM System i machine by entering the following command:

   `put ADAS400.SAVF` *library_name*`/ADAS400`

   where *library_name* is the name of the library where the save file ADAS400 exists.
5. After the transfer is complete, sign on to the IBM System i machine and restore the example library `ADAS400` by entering the following command:

   `RSTLIB SAVLIB(ADAS400) DEV(*SAVF) SAVF(`*library_name*`/ADAS400)`

   where *library_name* is name of the library where the save file ADAS400 exists.

The example library is compatible with IBM i operating system version V5R4 or higher.

Setting Up the Environment for TIBCO ActiveMatrix

If the environment for TIBCO ActiveMatrix has not been configured, you need first to complete the following tasks.

Task A Start TIBCO Enterprise Message Service Server

To start TIBCO Enterprise Message Service Server:

1. Go to the *EMS_HOME*`\bin` directory.
2. Run the following command at the command prompt:

   `tibemsd.exe`

Task B Start TIBCO ActiveMatrix Administrator Server

1. Start HSQLDB.

   — Run *AMX_HOME*`\amx\hsqldb\bin\amx-db.exe`.

   Or

   — (Windows only) Select **start > All Programs > TIBCO > Matrix_Home > TIBCO ActiveMatrix 2.2 > Start HSQLDB Server**.

2. Create a TIBCO ActiveMatrix Administrator Server with Administrator Server Creation Wizard if it is your first time to configure the environment.
3. Start Management Daemon.

   — Run *AMX_HOME*`\managementdaemon\2.0\bin\managementdaemon.exe`.

   Or

   — (Windows only) Select **start > All Programs > TIBCO > Matrix_Home > TIBCO Management Daemon 2.0 > Management Daemon**.

4. Start TIBCO ActiveMatrix Administrator.

   Run *AMX_HOME*`\amxadministrator\2.2\bin\amx_`*admin*`.bat`.

If ActiveMatrix Service Grid is installed within TIBCO Runtime Agent, the *AMX_HOME* should be *TIBCO_HOME*.

5. Start the TIBCO ActiveMatrix Administrator Server in a web browser:

   — Run *AMX_HOME*`\amxadministrator\data\bootstrap.html`.

Or

— (Windows only) Select **start > All Programs > TIBCO > Matrix_Home > TIBCO ActiveMatrix 2.2 > Administrator Servers**.

After starting the TIBCO ActiveMatrix Administrator Server, create your admin environment and node either with Administrator Server Creation Wizard or in a web browser.

The following tasks demonstrate how to do that in a web browser.

Task C Create an Admin Environment

To create an environment in a web browser:

1. Select the **Configure Enterprise Assets** perspective.
2. Click **New** to create a new Environment. Enter the general information. Click **Save**.
3. Select **Messaging Bus** under the environment that is just created.
4. Click **Add** to create a Messaging Server.
5. Select **Machines** under the environment.
6. Click the **Edit** button above the machines list. In the Enabled column of the machines list, select **Yes** for the machine that is to be associated with the environment. Click **Save**.

   Ensure that the following types of containers are listed in the Machines area: `SOAP`, `JMS`, `Java`, `.NET`, `Adapter for IBMi`, `Adapter`, `BusinessWorks`, and `Service Bus Mediation`.

Task D Create Node for the Environment

To create a node for the environment in a web browser:

1. Click **Perspective** and select **Configure an Environment**.
2. Select the environment that is created in the previous task from the Environment drop-down list.
3. Click **New** to create a node. Enter information for the node. The highlighted text boxes are mandatory.
4. Verify the container for the node.
5. Click **Install** to install the node into the environment.

Chapter 2 Publisher and RPC Client Example

This example demonstrates a Publisher that publishes a message from an application program running on the IBM System i machine; and a RPC Client that sends a request from an application running on an IBM System i machine and receives a reply with the requested information.

Topics

- Example Description, page 8
- Setting Up the Example, page 9
- Running the Example, page 15
- Expected Results, page 18

Example Description

This example consists of a Publisher named `CustomerMessage_Publisher` and a RPC Client named `CustomerInformation_client`.

Publisher

In the Publisher example:

1. An application program named `PUBLSH_PGM` sets the values of message data structure elements and invokes a data queue writer program named `PUBLISHER` on an IBM System i machine.
2. The data queue writer program named `PUBLISHER` writes the message to the outbound data queue.
3. The adapter gets the message and the Publisher publishes the message to the TIBCO environment.
4. The TIBCO ActiveMatrix BusinessWorks process receives the message.

RPC Client

In the RPC Client example:

1. An application program running on an IBM System i machine invokes the data queue writer/reader program.
2. The data queue writer/reader program writes a message with key information to the outbound data queue.
3. The RPC Client gets the message and sends it to the TIBCO environment.
4. The TIBCO ActiveMatrix BusinessWorks process receives the message and sends the requested customer information back.
5. The RPC Client receives the response and writes the response to the inbound data queue with the message key.
6. The data queue writer/reader program receives the response and sets the return parameter values.

Setting Up the Example

To set up the example, you need to complete the following configurations:

- Task A, Import the Projects, page 9
- Task B, Customize the Adapter Project, page 9
- Task C, Create ILE RPG Application to Publish Messages, page 11
- Task D, Create the Request-Response ILE RPG Program, page 11
- Task E, Configure the TIBCO ActiveMatrix BusinessWorks Processes, page 12
- Task F, Refresh Service Assembly and Build Archive, page 14

Task A Import the Projects

1. Start TIBCO Business Studio.
2. From the File menu, select **Import**.
3. In the Import dialog, select **General > Existing Projects into Workspace**. Click **Next**.
4. In the Import dialog, check the **Select archive file** radio button, browse to select the example archive file *AMX_HOME*`\extensions\adas400se\6.0\samples\IBMi_Adapter_Service_Engine_Samples.zip`.
5. In the Projects pane, select the Adapter Projects named `IBMiAdapterOutboundSample`, `IBMiAdapterOutboundBWSample`, and the SOA Project named `IBMiAdapterOutboundSOASample`.
6. Ensure that the **Copy projects into workspace** checkbox is selected.
7. Click **Finish**.

Task B Customize the Adapter Project

After importing the projects, you need to change the connection parameters as required.

1. Customize the connection parameters.
   a. In the Project Explorer, expand **IBMiAdapterOutboundSample** > **Shared Resources**.
   b. Double-click **IBMi_Connection_Configuration.sharedadas400conn** to launch the IBM i Connection Shared Resource Editor.
   c. In the IBM i Connection Shared Resource Editor, change the connection parameters as required. Click **Test Connection**.
   d. Save the changes.
2. In the Project Explorer, expand **IBMiAdapterOutboundSample**, and then double-click **IBMi_Adapter_Configuration.adas400model** to launch the editor of the adapter instance.
3. Click the **Adapter Services** tab.
4. Click the adapter service named `CustomerMessage_Publisher`, which is a Publisher. Check the configuration of the adapter service.
   — Class Reference: `CustomerMessage_Publisher`.
   — Data Queue for Publish: `ADAS400/ADAS400PUB`
   — Generate in Source File: `ADAS400/ADAS400SRC`
   — Generate Name of Member: `DS_CUSTMG`

   Click **UPLOAD** to generate ILE RPG source for the schema on the IBM System i machine.
5. Click the adapter service named `CustomerInformation_Client`, which is a RPC Client. Check the configuration of the adapter service.
   — Class Reference: `CustomerInformation_Client`.
   — Services Type: Outbound Message
   — Request Queue: `ADAS400/ADAS400RQS`
   — Reply Queue: `ADAS400/ADAS400RPY`
   — Generate in Source File: `ADAS400/ADAS400SRC`
   — Generate Name of Member: `DS_CUSTINF`

   Click **UPLOAD** to generate ILE RPG source for the schema on the IBM System i machine.
6. Save the project.

7. Regenerate WSDL file for the adapter.
   a. In the Project Explorer, right-click the adapter instance named `IBMi_Adapter_Configuration.adas400model`, and then select **Generate Adapter WSDL** from the shortcut menu.
   b. In the Target Project dialog, select the SOA project named `IBMiAdapterOutboundSOASample`, click **OK**.
   c. Click **OK** when being prompted to overwrite the existing resource.

Task C Create ILE RPG Application to Publish Messages

An application program on the IBM System i machine will use the ILE RPG schema source to set the values of various message fields, as well as publish the message by calling the data queue writer program `PUBLISHER`, which is also generated by the adapter palette.

A sample application program named `PUBLSH_PGM` is provided in the examples library `ADAS400` on the IBM System i machine, as follows:

```
*************** Beginning of data ************************************
D/Copy ADAS400/ADAS400SRC,DS_CUSTMSG
*-------------------------------------------------------------------
*  Move values to message variables
C                   Eval      Entry.Customer_Id = 4245
C                   Eval      Entry.Customer_Name = 'Test from HAWK400'
C                   Eval      Contact_Info.Tel_No = '1-800-GET-TIBCO'
C                   Eval      Contact_Info.Email = 'getTIBCO@tibco.com'
C                   Eval      Address.Line1 = '3303 HillView Ave.'
C                   Eval      Address.Line2 = *Blanks
C                   Eval      Address.City = 'Palo Alto'
C                   Eval      Address.State = 'CA'
C                   Eval      Address.ZipCode = 94304

*  Call 'PUBLISHER' program generated by TIBCO Adapter for IBM AS/400
*  message to external applications
C                   Call      'PUBLISHER'
C                   Parm                    MessageHeader
C                   Parm                    EntryBuffer
*
C                   Move      *On           *InLR
****************** End of data ***************************************
```

Task D Create the Request-Response ILE RPG Program

An application program on the IBM System will use the ILE RPG schema source to set the values of various message fields, send the message, and receive a reply message by calling the data queue writer/reader program `RPCCLIENT`, which is also generated by the adapter palette.

A sample application program REQRPY_PGM is available in the adapter example library ADAS400 on the IBM System i machine, as follows:

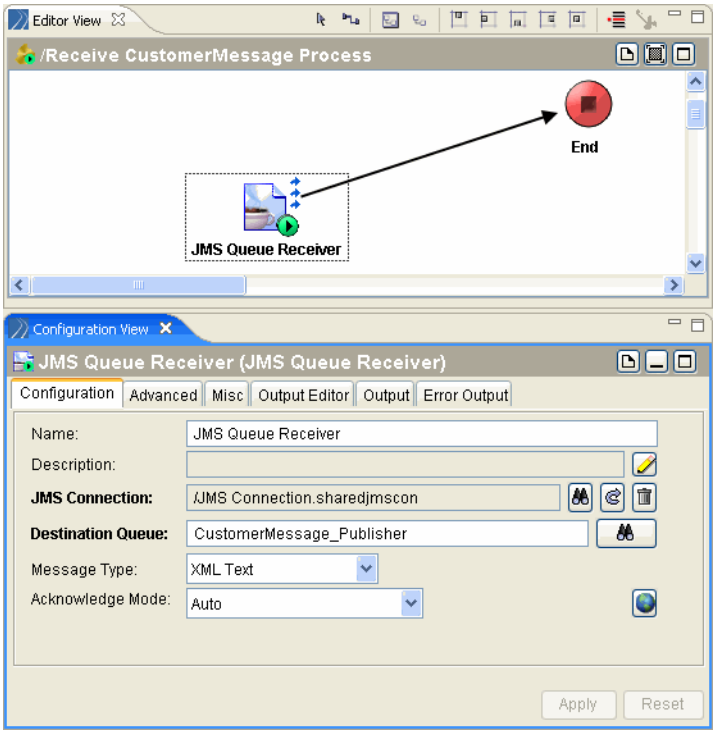
```
*************** Beginning of data ************************************
     D/Copy ADAS400/ADAS400SRC,DS_CUSTINF
*-----------------------------------------------------------------------------*
*  Move customer number to message data structure
*
C           Eval         Request.Customer_Id = 1234
*  Call 'RPCCLIENT' program generated by TIBCO Adapter for IBM AS/400 to
*  retrieve customer details from external systems
C           Call        'RPCCLIENT'
C           Parm         MessageHeader
C           Parm         KeyBuffer
C           Parm         RequestBuffer
C           Parm         ResponseBuffer
*  Use retrieved values here. Address values as -
*  Response.Customer_Name
*  Contact_Information.Telephone_Number
*  Mailing_Address.Address_Line_1
*
    :
    :
C           Move        *On          *InLR
***************** End of data ***************************************
```

Task E Configure the TIBCO ActiveMatrix BusinessWorks Processes

Two TIBCO ActiveMatrix BusinessWorks processes are defined in the project named IBMiAdapterOutboundBWSample. In most cases, you need not to change the configuration of the processes.

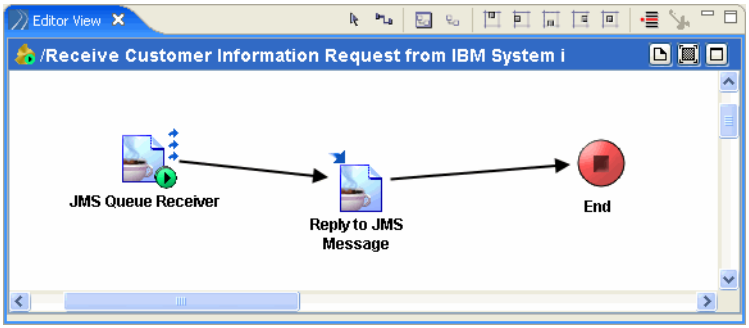
1. In the Project Explorer, select the project named IBMiAdapterOutboundBWSample, and then double-click the process file named Receive CustomerMessage Process.
2. In the Editor view, select the activity named JMS Queue Receiver, check the configuration.

Figure 1 Receive CustomerMessage Process



3. In the Project panel, double-click the process file named `Receive Customer Information Request from IBM System i`.

Figure 2 Receive Customer Information Request from IBM System i



4. In the Editor View, select the activity named `JMS Queue Receiver`. Ensure that the Destination Queue field is set to CustomerInformation_Client.
5. In the Editor View, select the activity named `Reply to JMS Message`. Ensure that the Reply To field is set to `JMS Queue Receiver`.

   Click the **Input** tab in the Configuration panel. The values for various fields in the message that will be sent back to the request application are specified here.

Task F Refresh Service Assembly and Build Archive

1. In the Project Explorer, select the SOA project named `IBMiAdapterOutboundSOASample`, and then expand the folder **Deployment Packages**.
2. Right-click the service assembly file `IBMiAdapterOutboundSOASample.saf`, and then select **Refresh Service Assembly** from the shortcut menu.
3. Right-click the `IBMiAdapterOutboundSOASample.saf` file, and then select **Build Archive** from the shortcut menu.
4. Click **Yes** when being prompted to overwrite the existing resource.

Running the Example

To run the example, complete the following tasks:

- Task A, Upload Service Assembly Archive, page 15
- Task B, Configure the Service Unit, page 15
- Task C, Start Node and Install Shared Resources, page 16
- Task D, Deploy and Start the Service Assembly, page 16
- Task E, Run the Publisher Example, page 17
- Task F, Run the RPC Client Example, page 17

Task A Upload Service Assembly Archive

Before proceeding, ensure that the HSQLDB, Management Daemon, TIBCO ActiveMatrix Administrator, and TIBCO ActiveMatrix Administrator Server in a web browser are running.

In TIBCO ActiveMatrix Administrator:

1. Select the **Deploy to an Environment** perspective.
2. Choose the Environment, and then click **Upload Service Assembly**.
3. Enter a name in the Name field.
4. Click **Browse** to select the archive file named `IBMiAdapterOutboundSOASample.zip` that is generated in TIBCO Business Studio.
5. Select the **Import Shared Resource Definitions** checkbox to create the shared resources definitions.
6. Click **OK**.

Task B Configure the Service Unit

1. In the Service Assemblies table, select the service assembly.

2. Click the **Service Units** button. Do the following steps for the service units listed in the Service Units table:
   a. Select a service unit.
   b. In the Node Mapping tab, click **Edit**, select the node that you want to deploy the service unit to in the Available Nodes list, and then click the right arrow button. The node will appear in the Mapped Nodes list.
   c. Click **Save**.

Task C Start Node and Install Shared Resources

1. Select **Configure an Environment** in the Perspective drop-down list.
2. In the Nodes table, select the appropriate node, and then click **Start**.
3. Select **Shared Resources**.
4. Click **Edit**. Enable the corresponding shared resource.
5. On the top of the Shared Resources table, click **Install**.

You can also start the node from the command line by running *AMX_HOME*`\amx\data\`*environment*`\`*node*`\bin\`*environment_node*`.exe`. In such a case, enable the shared resource and it will be installed once the node starts.

Task D Deploy and Start the Service Assembly

1. Select **Deploy to an Environment** in the Perspective drop-down list.
2. Select the service assembly, and then click **Deploy**.
3. After the status of the service assembly changes to `Deployed`. Select the service assembly and click **Start**.

To deploy the service assembly from the console of the node, enter the following command:

`dsa` *Service_Assembly_Name*

To start the service assembly from the console of the node, enter the following command:

`stsa` *Service_Assembly_Name*

Task E Run the Publisher Example

1. In TIBCO Business Studio, run the TIBCO ActiveMatrix BusinessWorks Process for the Publisher.
   a. From the Run menu, select **Run**.
   b. In the Run dialog, double-click **TIBCO Designer Application** from the filter list.
   c. Select the TIBCO Designer Project named `IBMiAdapterOutboundBWSample`.
   d. Select the process named `Receive CustomerMessage Process`.
   e. Click **Apply**. Click **Run**.
   f. In the Select Processes to Load dialog, click **Load & Start Current**.
2. Publish a message.

   Run the following command from the command line of the IBM System i machine:

   ```
   CALL ADAS400/PUBLSH_PGM
   ```

Task F Run the RPC Client Example

1. In TIBCO Business Studio, run the TIBCO ActiveMatrix BusinessWorks Process for the RPC Client.
   a. From the Run menu, select **Run**.
   b. In the Run dialog, double-click **TIBCO Designer Application** from the filter list.
   c. Select the TIBCO Designer Project named `IBMiAdapterOutboundBWSample`.
   d. Select the process named `Receive Customer Information Request from IBM System i`.
   e. Click **Apply**. Click **Run**.
   f. In the Select Processes to Load dialog, click **Load & Start Current**.
2. Start the Request-Response ILE RPG program.

   To send a request and receive a reply, invoke the ILE RPG application program. Run the following command from the command line of the IBM System i machine:

   ```
   CALL ADAS400/REQRPY_PGM
   ```

Expected Results

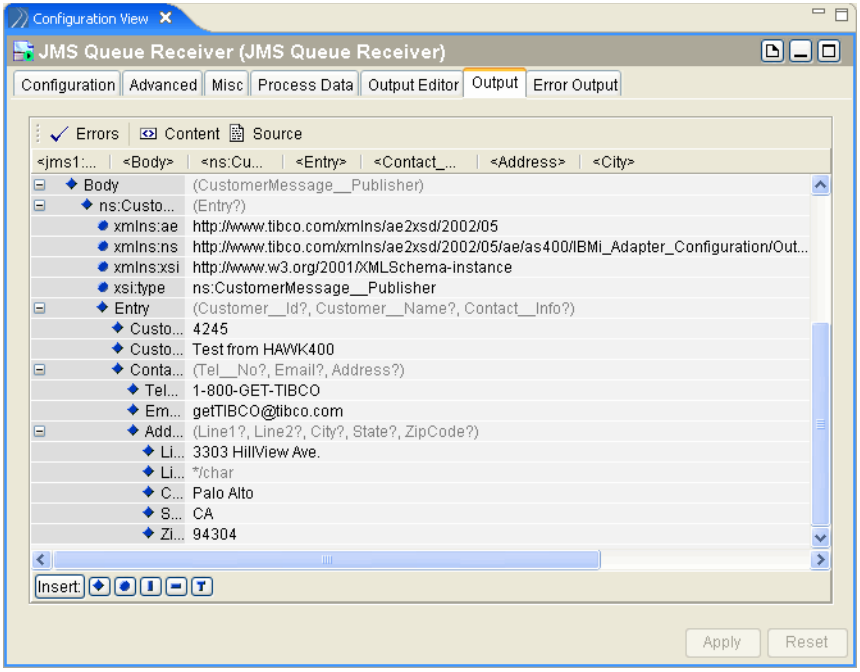
Publisher

After starting the Publisher ILE RPG program:

1. A message is sent to the output queue on the IBM System i machine.
2. The adapter gets the message from the queue and sends it to the TIBCO environment.
3. The `JMS Queue Receiver` activity receives the message.

You can view the message in the Output tab of the `JMS Queue Receiver` activity.

Figure 3 Published Message



RPC Client

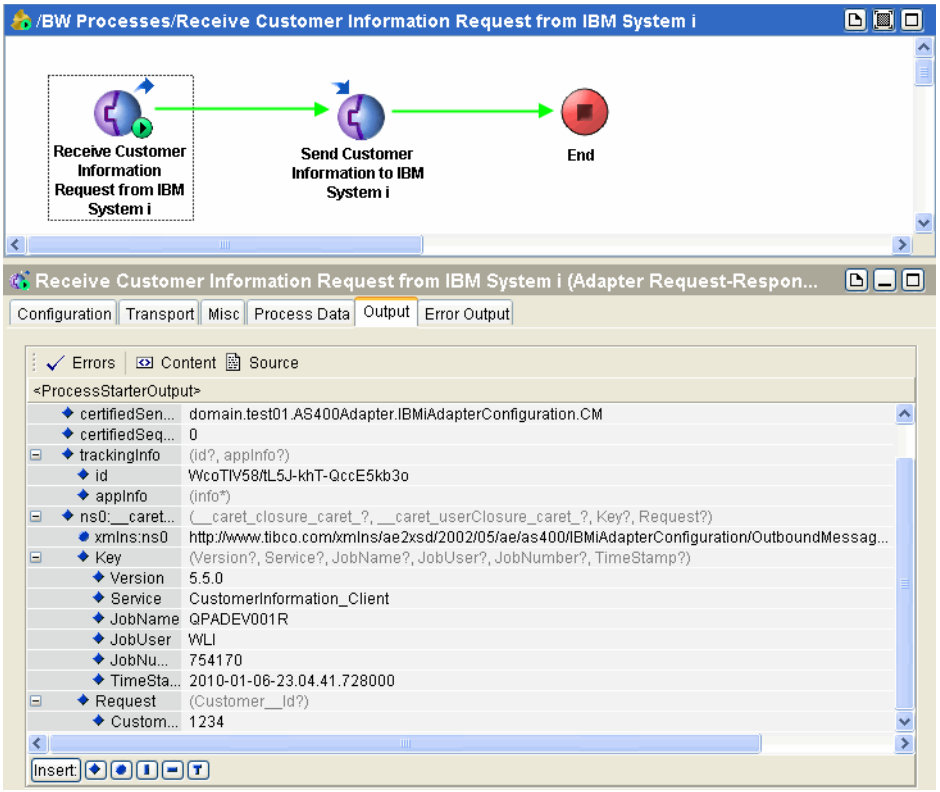
After starting the Request-Response ILE RPG program:

1. A request message is sent to the output queue on the IBM System i machine.
2. The adapter gets the request message and sends it to the TIBCO environment.

3. The `JMS Queue Receiver` activity receives the request message and processes it.
4. The `Reply to JMS Message` activity replies to the received message after setting required data values.

You can view the request message in the Output tab of the `JMS Queue Receiver` activity.

Figure 4 Request Message



Chapter 3 Subscriber (Program Call) and RPC Server (Data Queue Write) Example

This example demonstrates a Subscriber that calls an application program on an IBM System i machine to update a customer record; and explains how to configure the adapter to write a message to a data queue on an IBM System i machine in synchronous mode.

Topics

- Example Description, page 22
- Setting Up the Example, page 23
- Running the Example, page 29
- Expected Results, page 32

Example Description

This example consists of a Subscriber named `Customer_Update_Subscriber` and a RPC Server named `Customer_Inquiry_Server`.

Subscriber (Program Call)

The Subscriber example explains how to configure the adapter for calling an application program (`*PGM` type object) on the IBM System i machine in asynchronous mode.

In the Subscriber example:

1. A process for program invocation starts in TIBCO ActiveMatrix BusinessWorks.
2. Subscription Service receives the request from the process and asynchronously invokes an application running on an IBM System i machine.

RPC Server (Data Queue Write)

The RPC Server example explains how to configure the adapter to write a message to a data queue on an IBM System i machine in synchronous mode.

In the RPC Server example:

1. The TIBCO ActiveMatrix BusinessWorks process sends a request message to query a customer record.
2. The RPC server receives the request, writes a message to a request data queue on the IBM System i machine.
3. The RPC Server receives a reply message from another data queue on an IBM System i machine.

Setting Up the Example

To set up the example, you need to complete the following configurations.

Task A Import the Projects

1. Start TIBCO Business Studio.
2. From the File menu, select **Import**.
3. In the Import dialog, select **General > Existing Projects into Workspace**. Click **Next**.
4. In the Import dialog, check the **Select archive file** radio button, browse to select the example archive file *AMX_HOME*`\extensions\adas400se\6.0\samples\IBMi_Adapter_Service_Engine_Samples.zip`.
5. In the Projects pane, select the Adapter Projects named `IBMiAdapterInboundSample`, `IBMiAdapterInboundBWSample`, and the SOA Project named `IBMiAdapterInboundSOASample`.
6. Ensure that the **Copy projects into workspace** checkbox is selected.
7. Click **Finish**.

Task B Customize the Adapter Project

After importing the projects, you need to change the connection parameters as required.

1. Customize the connection parameters.
   a. In the Project Explorer, expand **IBMiAdapterInboundSample** > **Shared Resources**.
   b. Double-click **IBMi_Connection_Configuration.sharedadas400conn** to launch the IBM i Connection Shared Resource Editor.
   c. In the IBM i Connection Shared Resource Editor, change the connection parameters as required. Click **Test Connection**.
   d. Save the changes.
2. In the Project Explorer, expand **IBMiAdapterInboundSample**, and then double-click **IBMi_Adapter_Configuration.adas400model** to launch the editor of the adapter instance.
3. Click the **Adapter Services** tab.

4. Click the adapter service named `Customer_Update_Subscriber`, which is a Subscriber. Check the configuration of the adapter service.
   - Class Reference: `Customer_Update_Subscriber`.
   - Services Type: Program Call
   - Program Object: `ADAS400/CUSTMR_UPD`
   - Source File: `ADAS400/ADAS400SRC`
   - Source Member: `CUSTMR_UPD`

     The Subscriber is configured for the `CUSTMR_UPD` program in the ADAS400 library. `CUSTMR_UPD` is an ILE RPG program and the source information cannot be retrieved from `*PGM` object.
5. Click the adapter service named `Customer_Inquiry_Server`, which is a RPC Server. Check the configuration of the adapter service.
   - Class Reference: `Customer_Inquiry_Server`.
   - Services Type: Data Queue Write
   - Request Queue: `ADAS400/ADAS400SRQ`
   - Reply Data Queue: `ADAS400/ADAS400SRY`
   - Generate in Source File: `ADAS400/ADAS400SRC`
   - Generate Name of Member: `DS_CUSTINQ`

   Click **UPLOAD** to generate ILE RPG source for the schema on the IBM System i machine.
6. Save the project.

Task C Create the Request-Response ILE RPG Program

For the RPC Server, an application program on the IBM System i machine will use the ILE RPG schema source to receive a request, to set the values of various message fields in the reply message, and to write a reply message to the reply data queue.

A sample application program `CUSTMR_INQ` is available in the adapter example library `ADAS400` on the IBM System i machine, as follows:

```
*************** Beginning of data ************************************
D/Copy ADAS400/ADAS400SRC,DS_CUSTINQ

 *-----------------------------------------------------------------------------*
 *  Call 'RPCSERVER' program generated by TIBCO Adapter for IBM AS/400 to
 * receive a request from a 'Data Queue Write' type request-response service
 *
 C                Call    'RPCSERVER'
```

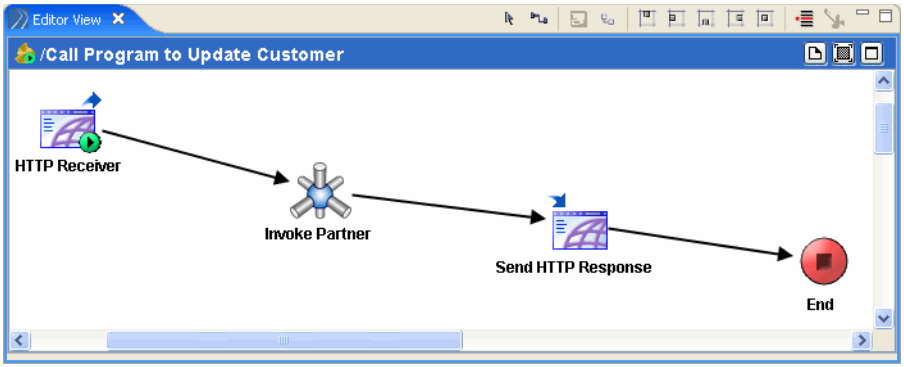
```
C                Parm      KeyBuffer
C                Parm      MessageHeader
C                Parm      RequestBuffer

*   Set values to retrieved message fields here. Address variables as -
*   Customer_Inquiry_Server.Customer_Name
*   Customer_Address.Address_Line_1
                        :
                        :
*   Call 'RPCSERVER' program generated by TIBCO Adapter for IBM AS/400 to
*   send reply for received request back to request-response service of
*   'Data Queue Write' type. In this call, data queue key will have value
*   that was set when request message was read.
*
C                Call     'RPCSERVER'
C                Parm      KeyBuffer
C                Parm      MessageHeader
C                Parm      RequestBuffer
*
C                Move       *On            *InLR
****************** End of data ****************************************
```

Task D Configure the TIBCO ActiveMatrix BusinessWorks Process

Two TIBCO ActiveMatrix BusinessWorks processes are defined in the project named `IBMiAdapterInboundBWSample`. In most cases, you need not to change the configuration of the processes.

1. In the Project Explorer, select the project named `IBMiAdapterInboundBWSample`, and then double-click the process file named `Call Program to Update Customer`.

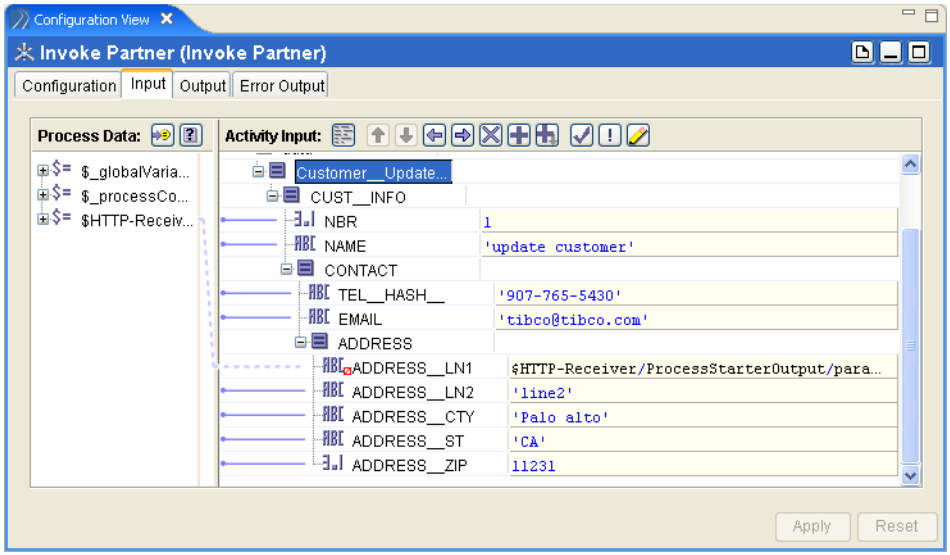
Figure 5 Call Program to Update Customer



2. In the Editor view, select the activity named `Invoke Partner`, check the configuration.

The customer row to be updated is identified by the field NBR, which is the customer number.

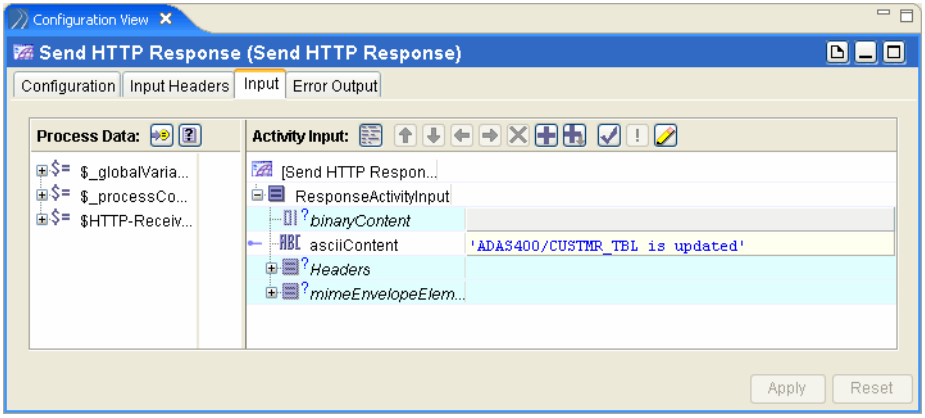
Figure 6 Invoke Partner



3. In the Editor view, select the activity named `Send HTTP Response`, check the configuration.

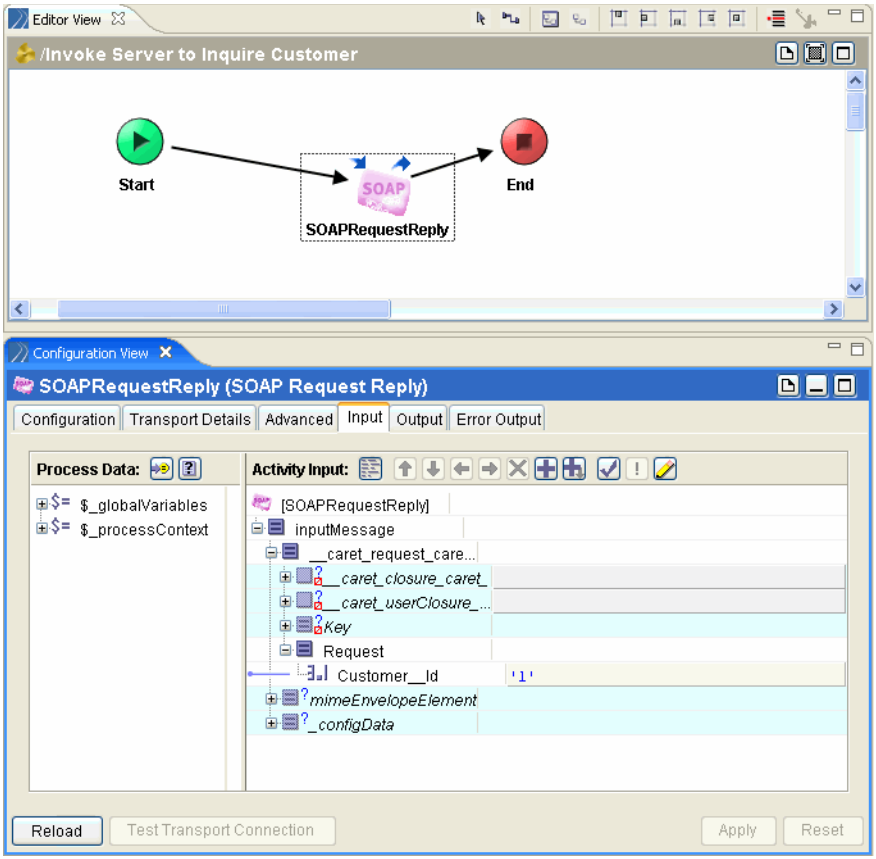
   The HTTP response message is defined in the field `asciiContent`.

Figure 7 Send HTTP Response



4. In the Project panel, double-click the process file named `Invoke Server to Inquire Customer`.
5. In the Editor View, select the activity named `SOAPRequestReply`. Click the **Input** tab in the Configuration View. The values for the schema fields are specified here.

Figure 8 Invoke Server to Inquire Customer



Task E Update the SOA Project

The adapter project being customized, you need to update resources in the SOA project.

1. Regenerate WSDL file for the adapter.
   a. In the Project Explorer, right-click the adapter instance named `IBMi_Adapter_Configuration.adas400model`, and then select **Generate Adapter WSDL** from the shortcut menu.
   b. In the Target Project dialog, select the SOA project named `IBMiAdapterInboundSOASample`, click **OK**.
   c. Click **OK** when being prompted to overwrite the existing resource.
2. Regenerate WSDL file for the SOAP composite service.
   a. In the Project Explorer, expand **IBMiAdapterInboundSample** > **Composites**, and then double-click `IBMiAdapterInboundSOAsample.composite`.
   b. Delete the existing WSDL file from the `Composites` folder.
   c. In the Composite Editor, select the SOAP service named `RPCService` on the canvas, click the **Binding** tab of the Properties view.
   d. Click the **Generate WSDL** button to generate the WSDL file that is used by the external client to access the service within ActiveMatrix.
3. Refresh Service Assembly and Build Archive.
   a. In the Project Explorer, expand **IBMiAdapterInboundSample** > **Deployment Packages**.
   b. Right-click the service assembly file `IBMiAdapterInboundSOASample.saf`, and then select **Refresh Service Assembly** from the shortcut menu.
   c. Right-click the `IBMiAdapterInboundSOASample.saf` file, and then select **Build Archive** from the shortcut menu.
   d. Click **Yes** when prompted to overwrite the existing resource.

Running the Example

To run the example, complete the following tasks:

- Task A, Upload Service Assembly Archive, page 29
- Task B, Configure the Service Unit, page 29
- Task C, Start Node and Install Shared Resources, page 30
- Task D, Deploy and Start the Service Assembly, page 30
- Task E, Run the Subscriber Example, page 31
- Task F, Run the RPC Server Example, page 31

Task A Upload Service Assembly Archive

Before proceeding, ensure that the HSQLDB, Management Daemon, TIBCO ActiveMatrix Administrator, and TIBCO ActiveMatrix Administrator Server in a web browser are running.

In TIBCO ActiveMatrix Administrator:

1. Select the **Deploy to an Environment** perspective.
2. Choose the Environment, and then click **Upload Service Assembly**.
3. Enter a name in the Name field.
4. Click **Browse** to select the archive file named `IBMiAdapterInboundSOASample.zip` that is generated in TIBCO Business Studio.
5. Select the **Import Shared Resource Definitions** checkbox to create the shared resources definitions.
6. Click **OK**.

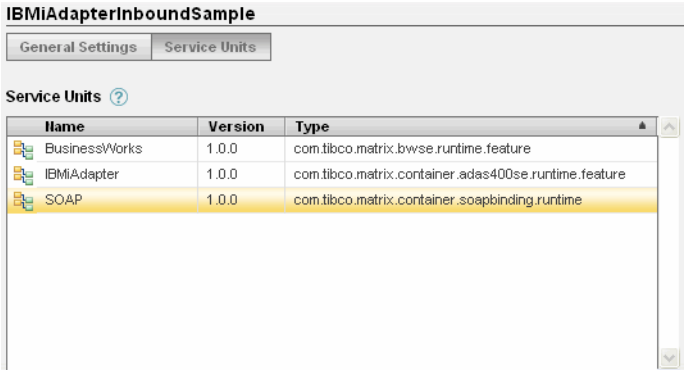
Task B Configure the Service Unit

There are three Service Units for this example: BusinessWorks, IBMiAdapter, and SOAP.

1. In the Service Assemblies table, select the service assembly.

2. Click the **Service Units** button. Do the following steps for the service units listed in the Service Units table:
   a. Select a service unit.
   b. In the Node Mapping tab, click **Edit**, select the node that you want to deploy the service unit to in the Available Nodes list, and then click the right arrow button. The node will appear in the Mapped Nodes list.
   c. Click **Save**.

Figure 9 Service Units of the Inbound Sample



Task C Start Node and Install Shared Resources

1. Select **Configure an Environment** in the Perspective drop-down list.
2. In the Nodes table, select the appropriate node, and then click **Start**.
3. Select **Shared Resources**.
4. Click **Edit**. Enable the corresponding shared resource.
5. On the top of the Shared Resources table, click **Install**.

You can also start the node from the command line by running *AMX_HOME*`\amx\data\`*environment*`\`*node*`\bin\`*environment_node*`.exe`. In such a case, enable the shared resource and it will be installed once the node starts.

Task D Deploy and Start the Service Assembly

1. Select **Deploy to an Environment** in the Perspective drop-down list.
2. Select the service assembly, and then click **Deploy**.

3. After the status of the service assembly changes to `Deployed`. Select the service assembly and click **Start**.

   Ensure that all the three Service Units, BusinessWorks, IBMiAdapter, and SOAP, are started.

To deploy the service assembly from the console of the node, enter the following command:

`dsa` *Service_Assembly_Name*

To start the service assembly from the console of the node, enter the following command:

`stsa` *Service_Assembly_Name*

Task E Run the Subscriber Example

1. Launch a browser, for example, the Internet Explorer.
2. Enter the following URL to invoke the TIBCO ActiveMatrix BusinessWorks process:

   `http://localhost:8091/?ADDRESS_LN1=address_line1`

Task F Run the RPC Server Example

1. In TIBCO Business Studio, run the TIBCO ActiveMatrix BusinessWorks Process for the RPC Server.
   a. From the Run menu, select **Run**.
   b. In the Run dialog, double-click **TIBCO Designer Application** from the filter list.
   c. Select the TIBCO Designer Project named `IBMiAdapterInboundBWSample`.
   d. Select the process named `Invoke Server to Inquire Customer`.
   e. Click **Apply**. Click **Run**.
   f. In the Select Processes to Load dialog, click **Load & Start Current**.
2. Start the Request-Response ILE RPG program.

   To receive a request and send a reply, invoke the ILE RPG application program by entering the following command:

   `CALL ADAS400/CUSTMR_INQ`

Expected Results

Subscriber (Program Call)

When you issued the HTTP request from the browser, the process named `Call Program to Update Customer` is invoked and then the process called the `CUSTMR_UPD` program. The `CUSTMR_UPD` program updates an existing customer row in the table `CUSTMR_TBL` in library ADAS400.

After the Subscriber is invoked, you could see the following response information in the browser, "`ADAS400/CUSTMR_TBL is updated`".

To view the results of the activities:

1. Sign-on to the IBM System i machine where the program and table in this example are located.
2. Start an interactive SQL session and run the `SELECT` command on `CUSTMR_TBL` in the ADAS400 library. Run the following commands from the command line menu of the IBM System i machine.

   ```
   STRSQL
   SELECT * FROM ADAS400/CUSTMR_TBL
   ```

3. Confirm that the record got updated correctly with the values set in the TIBCO ActiveMatrix BusinessWorks process. Refer to Figure 6 on page 26 for the values.

RPC Server (Data Queue Write)

After starting the `Invoke Server to Inquire Customer` process:

1. A message is written to the request data queue on the IBM System i machine.
2. The ILE RPG program running on IBM System i machine reads the request message from the request data queue, sets values in the message, and writes a reply message to the reply data queue.
3. The adapter service reads the reply message from the reply data queue and sends it as a reply to the TIBCO ActiveMatrix BusinessWorks process.

You can view the reply message in the Output tab of the `SOAPRequestReply` activity.

Index

A

ADAS400_HOME x
AMX_HOME x

C

create node 5
customer support xiii
Customer_Update_Subscriber 22
CustomerInformation_client 8
CustomerMessage_Publisher 8

E

environment 5

N

Node 5

O

overview 2

P

prerequisites 3
Publisher 8
PUBLSH_PGM 8, 11

R

REQRPY_PGM 12
RPC Client 8
RPC Serve 22

S

SOAP 28
Subscriber 22
support, contacting xiii

T

technical support xiii
TIBCO_HOME x