

# **TIBCO ActiveMatrix® Adapter Service Engine for Lotus Notes**

## **Examples**

*Software Release 6.0  
November 2009*

## 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.

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 © 2008-2009 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information

# Contents

<b>Preface</b> .....	<b>v</b>
Related Documentation .....	vi
TIBCO ActiveMatrix Adapter Service Engine for Lotus Notes Documentation .....	vi
Other TIBCO Product Documentation .....	vii
Third-Party Documentation .....	vii
Typographical Conventions .....	viii
How to Contact TIBCO Support .....	x
 <b>Chapter 1 Introduction</b> .....	 <b>1</b>
Overview .....	2
Prerequisites .....	3
Setting Up the Environment for TIBCO ActiveMatrix .....	3
 <b>Chapter 2 Subscriber Binding with SOAP</b> .....	 <b>7</b>
Example Description .....	8
Setting Up the Example .....	9
Running the Example .....	12
Expected Results .....	14
 <b>Chapter 3 RPC Client Binding with JMS</b> .....	 <b>15</b>
Example Description .....	16
Setting Up the Example .....	17
Running the Example .....	21
Expected Results .....	24
 <b>Chapter 4 RPC Server with Dynamic Login</b> .....	 <b>25</b>
Example Description .....	26
Setting Up the Example .....	27
Running the Example .....	30
Expected Results .....	32

**Chapter 5 RPC Server Interacting with RPC Client . . . . . 33**

Example Description . . . . . 34

Setting Up the Example . . . . . 35

Running the Example . . . . . 39

Expected Results . . . . . 41

**Chapter 6 Publisher Binding with BWSE . . . . . 43**

Example Description . . . . . 44

Setting Up the Example . . . . . 45

Running the Example . . . . . 48

Expected Results . . . . . 50

**Chapter 7 RPC Server Binding with BWSE . . . . . 51**

Example Description . . . . . 52

Setting Up the Example . . . . . 53

Running the Example . . . . . 56

Expected Results . . . . . 58

**Chapter 8 Subscriber with Dynamic Login . . . . . 59**

Example Description . . . . . 60

Setting Up the Example . . . . . 61

Running the Example . . . . . 64

Expected Results . . . . . 66

**Chapter 9 EAR to SA Example . . . . . 67**

Example Description . . . . . 68

Setting Up the Example . . . . . 69

Running the Example . . . . . 71

Expected Results . . . . . 73

**Index . . . . . 75**

# Preface

This guide explains how to use the examples to further your understanding of TIBCO ActiveMatrix Adapter Service Engine for Lotus Notes.

## Topics

---

- [Related Documentation, page vi](#)
- [Typographical Conventions, page viii](#)
- [How to Contact TIBCO Support, page x](#)

## Related Documentation

---

This section lists documentation resources you may find useful.

### TIBCO ActiveMatrix Adapter Service Engine for Lotus Notes Documentation

The following documents form the TIBCO ActiveMatrix Adapter Service Engine for Lotus Notes documentation set:

- *TIBCO ActiveMatrix Adapter Service Engine for Lotus Notes Concepts* Read this manual before reading any other book in the documentation set. This manual describes the adapter, adapter features, and the applications with which the adapter interacts.
- *TIBCO ActiveMatrix Adapter Service Engine for Lotus Notes Installation* Read this manual for instructions on how to install and uninstall the adapter service engine on Windows and UNIX platforms.
- *TIBCO ActiveMatrix Adapter Service Engine for Lotus Notes Configuration and Deployment* Read this manual to learn how to create and configure adapter projects. Information on deploying adapter projects is also included.
- *TIBCO ActiveMatrix Adapter Service Engine for Lotus Notes Examples* Read this manual to work through the examples provided with the adapter service engine.
- *TIBCO ActiveMatrix Adapter Service Engine for Lotus Notes Release Notes* Read the release notes for information about new features, changed features, and open and closed issues.

Before TIBCO ActiveMatrix Adapter Service Engine for Lotus Notes can be installed and used, you must install the TIBCO ActiveMatrix Adapter for Lotus Notes. The following documents form the standalone adapter documentation set:

- *TIBCO ActiveMatrix Adapter for Lotus Notes Concepts* Read this manual before reading any other book in the documentation set. This manual describes the adapter, adapter features, and the applications with which the adapter interacts.
- *TIBCO ActiveMatrix Adapter for Lotus Notes Installation* Read this manual for instructions on how to install and uninstall the standalone adapter on Windows and UNIX platforms.
- *TIBCO ActiveMatrix Adapter for Lotus Notes Configuration and Deployment* Read this manual to learn how to create and configure standalone adapter projects. Information on deploying adapter projects is also included.

- *TIBCO ActiveMatrix Adapter for Lotus Notes Examples* Read this manual to work through the examples provided with the adapter.
- *TIBCO ActiveMatrix Adapter for Lotus Notes Release Notes* Read the release notes for information about new features, changed features, and open and closed issues.

## 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 following IBM Lotus Notes documentation:

- *Lotus Notes Help*
- *Lotus Domino Server Documentation*
- *Lotus Domino Designer Documentation*

# Typographical Conventions

The following typographical conventions are used in this manual.

Table 1 General Typographical Conventions

Convention	Use
<i>TIBCO_HOME</i> <i>ENV_HOME</i> <i>AMX_HOME</i>	<p>Many TIBCO products must be installed within the same home directory. This directory is referenced in documentation as <i>TIBCO_HOME</i>. The value of <i>TIBCO_HOME</i> depends on the operating system. For example, on Windows systems, the default value is C:\tibco.</p> <p>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 <i>ENV_HOME</i>. The value of <i>ENV_HOME</i> depends on the operating system. For example, on Windows systems the default value is C:\tibco.</p> <p>TIBCO ActiveMatrix installs into a directory within <i>TIBCO_HOME</i>. This directory is referenced in documentation as <i>AMX_HOME</i>. The value of <i>AMX_HOME</i> depends on the operating system. For example on Windows systems, the default value is C:\tibco\amx\.</p> <p>TIBCO ActiveMatrix Adapter Service Engine for Lotus Notes installs in a directory within <i>AMX_HOME</i>.</p>
code font	<p>Code font identifies commands, code examples, filenames, pathnames, and output displayed in a command window. For example:</p> <p>Use MyCommand to start the foo process.</p>
<b>bold code font</b>	<p>Bold code font is used in the following ways:</p> <p>In procedures, to indicate what a user types. For example: Type <b>admin</b>.</p> <p>In large code samples, to indicate the parts of the sample that are of particular interest.</p> <p>In command syntax, to indicate the default parameter for a command. For example, if no parameter is specified, MyCommand is enabled:</p> <p>MyCommand [<b>enable</b>   disable]</p>
<i>italic font</i>	<p>Italic font is used in the following ways:</p> <p>To indicate a document title. For example: See <i>TIBCO ActiveMatrix BusinessWorks Concepts</i>.</p> <p>To introduce new terms For example: A portal page may contain several portlets. <i>Portlets</i> are mini-applications that run in a portal.</p> <p>To indicate a variable in a command or code syntax that you must replace. For example: MyCommand <i>PathName</i></p>



Table 1 General Typographical Conventions (Cont'd)




Convention	Use
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
{ }	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**

The examples in this manual demonstrate how to deploy and run the adapter in the TIBCO ActiveMatrix Environment. Details about how to use the examples are explained in later chapters of this book.

### Topics

---

- [Overview, page 2](#)
- [Prerequisites, page 3](#)

## Overview

---

TIBCO ActiveMatrix Adapter Service Engine for Lotus Notes ships with the following examples:

- The example in [Chapter 2](#) demonstrates how to configure Subscription Service in TIBCO Business Studio, wire it with SOAP service, and then run it in TIBCO ActiveMatrix Administrator to perform an insert operation.
- The example in [Chapter 3](#) demonstrates how to configure RPC Client in TIBCO Business Studio, wire it with JMS service, and then run it in TIBCO ActiveMatrix Administrator to perform a query operation.
- The example in [Chapter 4](#) demonstrates how to configure RPC Server in TIBCO Business Studio, wire it with SOAP service, and then run it in TIBCO ActiveMatrix Administrator to perform an insert operation with dynamic login.
- The example in [Chapter 5](#) demonstrates how to configure the RPC Client to act as the client for RPC Server directly in TIBCO Business Studio.
- The example in [Chapter 6](#) demonstrates how to configure Publication Service in TIBCO Business Studio, wire it with BWSE service, and then run it in TIBCO ActiveMatrix Administrator.
- The example in [Chapter 7](#) demonstrates how to configure RPC Server in TIBCO Business Studio, wire it with BWSE service, and then run it in TIBCO ActiveMatrix Administrator.
- The example in [Chapter 8](#) demonstrates how to configure Subscription Service in TIBCO Business Studio, wire it with BWSE service, and then run it in TIBCO ActiveMatrix Administrator to perform an insert operation with dynamic login.
- The example in [Chapter 9](#) demonstrates how to convert an EAR file generated in TIBCO Designer to service assembly in TIBCO Business Studio, and then run it in TIBCO ActiveMatrix Administrator.

## Prerequisites

---

The adapter installation package includes sample Lotus Notes databases (.nsf files) with TIBCO templates for creating TIBCO enabled applications. Before you can work in the TIBCO provided databases, open the databases in Lotus Notes and upload the databases to Domino server. See *TIBCO ActiveMatrix Adapter Service Engine for Lotus Notes Configuration and Deployment* for details.

The .nsf database files are located at `ADLN_HOME/examples/LotusDatabase`.

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

## 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 > TIBCO ActiveMatrix 2.2 > Start HSQLDB Server**.
2. Create a TIBCO ActiveMatrix Administrator Server with Admin Server Creation Wizard if it is the first time for you 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 > 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 > 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 LotusNotes, Adapter, 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

# Subscriber Binding with SOAP

This chapter illustrates how to perform an insert operation with the Subscription Service.

## Topics

---

- [Example Description, page 8](#)
- [Setting Up the Example, page 9](#)
- [Running the Example, page 12](#)
- [Expected Results, page 14](#)

## Example Description

---

The example uses the `customer` form included in the `sample.nsf` database to perform an insert operation with a Subscription Service.

The adapter project is wired with SOAP service and runs in TIBCO ActiveMatrix Administrator. A TIBCO Designer project is predefined to include a TIBCO ActiveMatrix BusinessWorks process that is to trigger the insert operation.

The example archive file can be found in the following location:

`AMX_HOME\extensions\adlnse\6.0\samples\AMX_UIBase`

## Setting Up the Example

---



Before proceeding, ensure that the environment for TIBCO ActiveMatrix has been set up correctly. See [Setting Up the Environment for TIBCO ActiveMatrix on page 3](#) for more information about how to configure the environment.

To set up the example, complete the following tasks:

- [Task A, Set up Lotus Notes Application, page 9](#)
- [Task B, Import the Adapter Project and SOA Project, page 9](#)
- [Task C, Customize the Adapter Project, page 10](#)
- [Task D, Update the SOA Project, page 10](#)
- [Task E, Import the Designer Project, page 11](#)

### Task A Set up Lotus Notes Application

To run the examples:

1. Start Lotus Notes and click **Databases**.
2. Open the `sample.nsf` file from `ADLN_HOME/examples/LotusDatabase`, and upload it to Domino Server.

### Task B Import the Adapter Project and SOA Project

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\adlnse\6.0\samples\AMX_UIBase\SubscriberDemo.zip`.
5. Ensure that the Adapter Project `SubDemo-UI` and the SOA Project `SubDemo-SOA` have been selected in the Projects pane.
6. Select the **Copy projects into workspace** checkbox.
7. Click **Finish**.

### Task C Customize the Adapter Project

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

1. In the Project Explorer, expand `SubDemo-UI > defaultVars`, and then double-click `defaultVars.substvar` to open the editor for substitution variables.
2. Change values for substitution variables `adln.connection.dominoserver`, `adln.connection.username`, and `adln.connection.password`.
3. Save the configuration.
4. In the Project Explorer, expand `SubDemo-UI > Shared Resources`, and then double-click `LotusNotes Connection.shareInc` to open the editor for connection.
5. Click the **Test Connection** button to ensure that the adapter can connect to Lotus applications.

### Task D 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, expand `SubDemo-UI > Adapter Configurations`, right-click the adapter instance `LotusNotesAdapterConfiguration.adlnmodel`, and then select **Generate Adapter WSDL** from the shortcut menu.
  - b. In the Target Project dialog, select the SOA project `SubDemo-SOA`, click **OK**.
  - c. Click **OK** when prompted to overwrite the existing resource.
2. Regenerate WSDL file for the SOAP composite service.
  - a. In the Project Explorer, expand `SubDemo-SOA > Composites`, and then double-click `SubDemo-SOA.composite`.
  - b. Delete the existing WSDL file from the `Composites` folder.
  - c. In the Composite Editor, select the SOAP service 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 SubDemo-SOA > Deployment Packages.
  - b. Right-click the service assembly file SubDemo-SOA.saf, and then select **Refresh Service Assembly** from the shortcut menu.
  - c. Right-click the SubDemo-SOA.saf file, and then select **Build Archive** from the shortcut menu.
  - d. Click **Yes** when prompted to overwrite the existing resource.

### Task E Import the Designer Project

1. Unzip the example archive file.
2. Start TIBCO Designer and select **Open existing project**.
3. In the Multi-File Project tab, select SubDemoTrigger-BW for the Project Directory field. Click **OK**.
4. Delete the existing WSDL file from the wsdlFiles folder.
5. Import new WSDL files.
  - a. From the Project menu, select **Import Resources from File, Folder, URL**.
  - b. In the Import Resources or File dialog, select **File (.xsd, .xslt, .wsdl, .\*)** from the Format list.
  - c. Click the **Select a file from the file system** button next to the File field to locate the .wsdl file that is generated for the SOAP composite service.
  - d. In the Project panel, double-click the process Trigger\_SubInsert.
  - e. Select the **InsertAction** activity in the Design panel.
  - f. In the Configuration tab, click the **Browse resources** button next to the Service field. In the Select a Resource dialog, click the **By Location** tab, and then select the WSDL file that you imported earlier. Click **OK**.
6. Click **Apply** and save the project.

## Running the Example

---

To run the example, complete the following tasks:

- [Task A, Upload Service Assembly Archive, page 12](#)
- [Task B, Configure the Service Unit, page 12](#)
- [Task C, Start Node and Install Shared Resources, page 13](#)
- [Task D, Deploy and Start the Service Assembly, page 13](#)
- [Task E, Run the Process, page 13](#)

### 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 `SOA_SubDemo.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

This example adopts an ActiveMatrix Service SOAP. You need to install the corresponding Shared Resource to the node.

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 Process

In TIBCO Designer:

1. Click the **Tester** tab, and then click the **Start testing viewed process** button.
2. Ensure that the `Trigger_subInsert` process is selected.
3. Click **Load Selected**.

## Expected Results

---

A new document is created in the Customer form of the target database (`sample.nsf`) to which the Subscription Service is connected.



## Chapter 3 **RPC Client Binding with JMS**

This chapter illustrates how to perform a query operation with the RPC Client.

### Topics

---

- [Example Description, page 16](#)
- [Setting Up the Example, page 17](#)
- [Running the Example, page 21](#)
- [Expected Results, page 24](#)

## Example Description

---

The example uses the `customer` form included in the `sample.nsf` file to perform query operation.

The example works with a Request-Response Service.

1. The TIBCO ActiveMatrix BusinessWorks process sends a pre-defined request to its related Request-Response Service.
2. The Request-Response Service performs a query operation in Lotus Notes database, and returns the result to the TIBCO ActiveMatrix BusinessWorks process.

The example archive file can be found in the following location:

`AMX_HOME\extensions\adlnse\6.0\samples\AMX_UIBase`

## Setting Up the Example

---

To set up the example, complete the following tasks:

- [Task A, Set up Lotus Notes Application, page 17](#)
- [Task B, Create RV Agents, page 17](#)
- [Task C, Import the Adapter Project and SOA Project, page 18](#)
- [Task D, Customize the Adapter Project, page 18](#)
- [Task E, Create a Request Response Invocation Document, page 19](#)
- [Task F, Update the SOA Project, page 19](#)
- [Task G, Import the Designer Project, page 20](#)

### Task A Set up Lotus Notes Application

To run the examples:

1. Start Lotus Notes and click **Databases**.
2. Open the `sample.nsf` file from `ADLN_HOME/examples/LotusDatabase`, and upload it to Domino Server.
3. Select the `sample.nsf` file that is at the Domino Server. Enter values for the following field in the Setup Document:
  - Queue Db Server: IP address of the machine which hosts the database file.
  - Host Name: IP address of the machine on which the adapter is running.
  - Publish Subject: the same as the Subject to Listen to LotusNotes Events field value that is specified in the Adapter Services tab of the adapter instance. In this example, it is `trigger_rpcc`.
  - RV Batch File for RPC: the path of the RV batch file. For example, `C:\Program Files\lotus\notes\RvInitClient.bat`.
4. Save the changes.

### Task B Create RV Agents

Create a RV agent to trigger the RPC Client:

1. In Lotus Notes, select **Create > Agent**.
2. In the Agent dialog, enter a name for the agent. For example, `RvInitClientRequest`.

3. In the action panel, select **Imported Java** from the Run list box. Click **Import Class Files** to import the class file `InitiateClientRequest` from `ADLN_HOME/lib/agents/RV/rv_agents`.
4. Save the configuration.

### Task C Import the Adapter Project and SOA Project

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\adlnse\6.0\samples\AMX_UIBase\CustomerRequest.zip`.
5. Ensure that the Adapter Project `RpcClientRequest-UI` and the SOA Project `RpcClientRequest-SOA` have been selected in the Projects pane.
6. Select the **Copy projects into workspace** checkbox.
7. Click **Finish**.

### Task D Customize the Adapter Project

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

1. In the Project Explorer, expand `RpcClientRequest-UI > defaultVars`, and then double-click `defaultVars.substvar` to open the editor for substitution variables.
2. Change values for substitution variables `adln.connection.dominoserver`, `adln.connection.username`, and `adln.connection.password`.
3. Save the configuration.
4. In the Project Explorer, expand `RpcClientRequest-UI > Shared Resources`, and then double-click `LotusNotes Connection.shareInc` to open the editor for connection.
5. Click the **Test Connection** button to ensure that the adapter can connect to Lotus applications.

## Task E Create a Request Response Invocation Document

1. In the Project Explorer, expand `RpcClientRequest-UI > Adapter Configurations`, and then double-click the adapter instance `ADLN_RpccRequestConfiguration.adlnmodel` to open the editor.
2. Click the **Adapter Services** tab, configure a view document in the Request-Response Invocation Request Document Editor. [Figure 1](#) is a sample configuration.

Figure 1 Configure a Request Response Invocation Document

The screenshot shows the 'Request-Response Invocation Request Document Editor' window. It contains several fields and controls:

- Operation Type:** A dropdown menu set to 'query'.
- Formula:** A text box containing 'Form="customer"'.
- View Name:** An empty text box.
- Parameters:** An empty text box.
- Agent Name:** An empty text box.
- Result Customized:** A checked checkbox.
- Result Form Priority:** Two radio buttons, 'Client' (selected) and 'Server'.
- Result Form:** A text box containing 'customer' and a 'Select Form' button.
- Requested Fields:** A text box containing 'areacode4, accountname, ac' and a 'Get Fields' button.
- At the bottom, there are two buttons: 'Read Target' and 'Apply Change'.

3. Click **Apply Change**.  
A Request Response Invocation document named `rpcClient`, which is the same name as that of the adapter service configuration, is created in Lotus Notes.
4. Save the configuration.
5. Do the following steps in Lotus Notes:
  - a. Open the `sample.nsf` database.
  - b. Expand `TIB Administration\5. Defined Client Requests`.
  - c. Copy a new client request document, change the Request name to `rpcClient`, and change the Client Name to `rpcClientEndpoint`.
  - d. Save the changes.

## Task F 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, expand `RpcClientRequest-UI` > `Adapter Configurations`, right-click the adapter instance `ADLN_RpccRequestConfiguration.adlnmodel`, and then select **Generate Adapter WSDL** from the shortcut menu.
  - b. In the Target Project dialog, select the SOA project `RpcClientRequest-SOA`, click **OK**.
  - c. Click **OK** when prompted to overwrite the existing resource.
2. Refresh Service Assembly and Build Archive.
  - a. In the Project Explorer, expand `RpcClientRequest-SOA` > `Deployment Packages`.
  - b. Right-click the service assembly file `CustomerRequest-SOA.saf`, and then select **Refresh Service Assembly** from the shortcut menu.
  - c. Right-click the `CustomerRequest-SOA.saf` file, and then select **Build Archive** from the shortcut menu.
  - d. Click **Yes** when prompted to overwrite the existing resource.

### Task G Import the Designer Project

1. Unzip the example archive file.
2. Start TIBCO Designer and select **Open existing project**. In the Multi-File Project tab, select `ProcessClientRequest-BW` for the Project Directory field. Click **OK**.
3. Change the configuration as necessary. For example, provide values for the global variables `adln.connection.dominoserver`, `adln.connection.username`, and `adln.connection.password`.
4. In the Project panel, select `RequestHandler` > `LotusNotesAdapterConfiguration`, click **Test Connection** in the Design-time Connection tab to ensure that the adapter can connect to the Lotus application.
5. Select `rpcS` under the `Adapter Services` folder, and then click **Get Schema** in the Configuration tab.
6. In the Project panel, select the process `ProcessRequest`. In the Design panel, select the activity `Invoke an Adapter Request-Response Service`.
7. In the Configuration tab, click **Refresh Adapter Service**, and then click **Apply**.
8. Save the configuration.

## Running the Example

---

To run the example, complete the following tasks:

- [Task A, Upload Service Assembly Archive, page 21](#)
- [Task B, Configure the Service Unit, page 21](#)
- [Task C, Start Node and Install Shared Resources, page 22](#)
- [Task D, Deploy and Start the Service Assembly, page 22](#)
- [Task E, Start the Request-Response Service and Run the Process, page 23](#)
- [Task F, Run the Agent, page 23](#)

### 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 `CustomerRequest-SOA.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

This example adopts an ActiveMatrix Service JMS. You need to install the corresponding Shared Resource to the node.

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 Start the Request-Response Service and Run the Process

In TIBCO Designer:

1. From the Tools menu, select **Show Adapter Tester**.
2. Select the adapter instance named `LotusNotesAdapterConfiguration`.
3. Click the **Run Settings** tab. In the Working Directory field, enter a directory to place running files.
4. In the Adapter Executable field, select the executable. For example, `TIBCO ActiveMatrix Adapter for Lotus Notes 6.0 (adln.exe)`. Click **Apply**.
5. Click **Start**. To view the output messages, go to the **Console** tab.
6. Click the **Tester** tab, and then click the **Start testing viewed process** button.
7. Ensure that the process named `ProcessRequest` is selected.
8. Click **Load Selected**.

## Task F Run the Agent

In Lotus Notes:

1. Select the request document named `rpcClient` in the Request Response Invocation View.
2. From the Actions menu, select the RV agent that is created when setting up this example.
3. Click `6. Client Request Processing` under the TIB Administration folder. The status of the request should be `Processed`.

## Expected Results

---

Double-click the request document in the database `sample.nsf`, and the query result of the documents are displayed in the Return Data field.

## Chapter 4 **RPC Server with Dynamic Login**

This example shows how to use Request-Response Service to perform an insert operation with dynamic login.

### Topics

---

- [Example Description, page 26](#)
- [Setting Up the Example, page 27](#)
- [Running the Example, page 30](#)
- [Expected Results, page 32](#)

## Example Description

---

The example uses the `customer` form included in the `sample.nsf` file to perform Insert operation.

The example archive file can be found in the following location:

`AMX_HOME\extensions\adlnse\6.0\samples\AMX_UIBase`

## Setting Up the Example

---

To set up the example, complete the following tasks:

- [Task A, Set up Lotus Notes Application, page 27](#)
- [Task B, Import the Adapter Project and SOA Project, page 27](#)
- [Task C, Customize the Adapter Project, page 28](#)
- [Task D, Update the SOA Project, page 28](#)
- [Task E, Import the Designer Project, page 29](#)

### Task A Set up Lotus Notes Application

To run the examples:

1. Start Lotus Notes and click **Databases**.
2. Open the `sample.nsf` file from `ADLN_HOME/examples/LotusDatabase`, and upload it to Domino Server.
3. Grant access to the `sample.nsf` file to the users who are required to perform the login example.

### Task B Import the Adapter Project and SOA Project

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\adlnse\6.0\samples\AMX_UIBase\DynamicRpcserver.zip`.
5. Ensure that the Adapter Project `DynamicRpcserver-UI` and the SOA Project `DynamicRpcserver-SOA` have been selected in the Projects pane.
6. Select the **Copy projects into workspace** checkbox.
7. Click **Finish**.

### Task C Customize the Adapter Project

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

1. In the Project Explorer, expand `DynamicRpcs-UI > defaultVars`, and then double-click `defaultVars.substvar` to open the editor for substitution variables.
2. Change values for substitution variables `adln.connection.dominoserver`, `adln.connection.username`, and `adln.connection.password`.
3. Save the configuration.
4. In the Project Explorer, expand `DynamicRpcs-UI > Shared Resources`, and then double-click `LotusNotes Connection.shareInc` to open the editor for connection.
5. Click the **Test Connection** button to ensure that the adapter can connect to Lotus applications.

### Task D 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, expand `DynamicRpcs-UI > Adapter Configurations`, right-click the adapter instance `ADLN_DynamicRpcsConfiguration.adlnmodel`, and then select **Generate Adapter WSDL** from the shortcut menu.
  - b. In the Target Project dialog, select the SOA project `DynamicRpcs-SOA`, click **OK**.
  - c. Click **OK** when prompted to overwrite the existing resource.
2. Regenerate WSDL files for the SOAP composite services.
  - a. In the Project Explorer, expand `DynamicRpcs-SOA > Composites`, and then double-click the composite file `DynamicRpcs-SOA.composite`.
  - b. Delete the three existing WSDL files from the `Composites` folder.
  - c. In the Composite Editor, select a SOAP service 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.
  - e. Generate WSDL for the other two SOAP services.

3. Refresh Service Assembly and Build Archive.
  - a. In the Project Explorer, expand `DynamicRpcs-SOA > Deployment Packages`.
  - b. Right-click the service assembly file `DynamicRpcs-SOA.saf`, and then select **Refresh Service Assembly** from the shortcut menu.
  - c. Right-click the `DynamicRpcs-SOA.saf` file, and then select **Build Archive** from the shortcut menu.
  - d. Click **Yes** when prompted to overwrite the existing resource.

### Task E Import the Designer Project

1. Unzip the example archive file.
2. Start TIBCO Designer and select **Open existing project**.
3. In the Multi-File Project tab, select `DynamicRpcsTrigger-BW` for the Project Directory field. Click **OK**.
4. Delete the existing WSDL files from the `wsdlFiles` folder.
5. Import new WSDL files.
  - a. From the Project menu, select **Import Resources from File, Folder, URL**.
  - b. In the Import Resources or File dialog, select **File (.xsd, .xslt, .wsdl, .\*)** from the Format list.
  - c. Click the **Select a file from the file system** button next to the File field to locate the `.wsdl` file that is generated for the SOAP composite service.  
Ensure that you import all the three WSDL files.
  - d. In the Project panel, double-click the process `DynamicRpcs_Trigger`.
  - e. Select the **Login** activity in the Design panel.
  - f. In the Configuration tab, click the **Browse resources** button next to the Service field.
  - g. In the Select a Resource dialog, click the **By Location** tab, and then select the WSDL file that you imported earlier. Click **OK**.  
Select **Replace old import** when a message indicating that the namespace is imported already pops up.
  - h. Repeat [step e](#) through [step g](#) for the other two activities of the process.
6. Select the **Login** activity in the Design panel.
7. In the Input tab, change the values of the `username` field and the `password` field to login to Domino Server. Save the project.

## Running the Example

---

To run the example, complete the following tasks:

- [Task A, Upload Service Assembly Archive, page 30](#)
- [Task B, Configure the Service Unit, page 30](#)
- [Task C, Start Node and Install Shared Resources, page 31](#)
- [Task D, Deploy and Start the Service Assembly, page 31](#)
- [Task E, Run the Process, 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 `DynamicRpcs-SOA.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

This example adopts an ActiveMatrix Service JMS. You need to install the corresponding Shared Resource to the node.

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 Process

In TIBCO Designer:

1. Click the **Tester** tab, and then click the **Start testing viewed process** button.
2. Ensure that the `Dynamic_Trigger` process is selected.
3. Click **Load & Start Current**.

## Expected Results

---

- Once the insert operation has been performed, the TIBCO ActiveMatrix BusinessWorks process logs out the user who performed the operation.
- Open Lotus Notes and check the view `customer` in `sample.nsf` to which the Request-Response Service is connected. New documents are created.

## Chapter 5 **RPC Server Interacting with RPC Client**

This example explains how to configure the Request-Response Invocation Service to act as a client for the Request-Response Service directly.

### Topics

---

- [Example Description, page 34](#)
- [Setting Up the Example, page 35](#)
- [Running the Example, page 39](#)
- [Expected Results, page 41](#)

## Example Description

---

This example explains how to configure the Request-Response Invocation Service to act as a client for the Request-Response Service directly.



The Request-Response Invocation Service can act as a client for the Request-Response Service only for the operation type `query`.

## Setting Up the Example

---

To set up the example, complete the following tasks:

- [Task A, Set up Lotus Notes Application, page 35](#)
- [Task B, Create Agents, page 35](#)
- [Task C, Import the Adapter Project and SOA Project, page 36](#)
- [Task D, Customize the Adapter Project, page 36](#)
- [Task E, Create a Request Response Invocation Document, page 36](#)
- [Task F, Update the SOA Project, page 37](#)

### Task A Set up Lotus Notes Application

The `sample.nsf` database is used in this example.

1. Start Lotus Notes and click **Databases**.
2. Configure the `sample.nsf` database.
  - a. Enter values for the following fields in the Setup Document:
 

Queue Db Server: IP address of the machine that hosts the database file.

Host Name: IP address of the machine on which the adapter is running.

Port Number: should be the same as the Http Port to Listen to LotusNotes Events field value that is specified in the Adapter Options tab. In this example, it is 8081.
  - b. Save the changes.

### Task B Create Agents

1. In Lotus Notes, select **Create > Agents**.
2. In the Name field, enter a name that indicates the action to be taken.  
For example, `HTTPInitiateClientRequest`.
3. Close the **Agent** dialog.
4. In the Action panel, select **Imported Java** from the Run list box.
5. Click **Import Class Files**. Import the class file  
`InitiateClientRequestHttp.class` from `ADLN_HOME/lib/agents/HTTP`.
6. Click **OK**.

7. Save the configuration.

### Task C Import the Adapter Project and SOA Project

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\adlnse\6.0\samples\AMX_UIBase\ClientInteractServerEndpoint.zip`.
5. Ensure that the Adapter Project `ClientInteractServerEndpoint-UI` and the SOA Project `ClientInteractServerEndpoint-SOA` have been selected in the Projects pane.
6. Select the **Copy projects into workspace** checkbox.
7. Click **Finish**.

### Task D Customize the Adapter Project

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

1. In the Project Explorer, expand `ClientInteractServerEndpoint-UI > defaultVars`, and then double-click `defaultVars.substvar` to open the editor for substitution variables.
2. Change values for substitution variables `adln.connection.dominoserver`, `adln.connection.username`, and `adln.connection.password`.
3. Save the configuration.
4. In the Project Explorer, expand `ClientInteractServerEndpoint-UI > Shared Resources`, and then double-click `LotusNotesConnection.shareInc` to open the editor for connection.
5. Click the **Test Connection** button to ensure that the adapter can connect to Lotus applications.

### Task E Create a Request Response Invocation Document

1. In the Project Explorer, expand `ClientInteractServerEndpoint-UI > Adapter Configurations`, and then double-click the adapter instance `ADLN_RpccInteractWithRpcConfiguration.adlnmodel` to open the editor.

2. Click the **Adapter Services** tab, configure a view document in the Request-Response Invocation Request Document Editor. [Figure 2](#) is a sample configuration.

*Figure 2 Configure a Request Response Invocation Document*

**Request-Response Invocation Request Document Editor**

Operation Type:

Formula:

View Name:

Parameters:

Agent Name:

Result Customized: ☒

Result Form Priority: ☒ Client ☐ Server

Result Form:

Requested Fields:

3. Click **Apply Change**.

A Request Response Invocation document named `rpcClient`, which is the same as that of the adapter service configuration, is created in Lotus Notes.

4. Save the configuration.
5. Do the following steps in Lotus Notes:
  - a. Open the `sample.nsf` database.
  - b. Expand `TIB Administration\5. Defined Client Requests`.
  - c. Copy a new client request document, change the Request name to `rpcClient`, and change the Client Name to `rpcClientEndpoint`.

## Task F 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, expand `ClientInteractServerEndpoint-UI > Adapter Configurations`, right-click the adapter instance

`ADLN_RpccInteractWithRpcsConfiguration.adlnmodel`, and then select **Generate Adapter WSDL** from the shortcut menu.

- b. In the Target Project dialog, select the SOA project `ClientInteractServerEndpoint-SOA`, click **OK**.
  - c. Click **OK** when prompted to overwrite the existing resource.
2. Refresh Service Assembly and Build Archive.
- a. In the Project Explorer, expand `ClientInteractServerEndpoint-SOA > Deployment Packages`.
  - b. Right-click the service assembly file `ClientInteractServerEndpoint-SOA.saf`, and then select **Refresh Service Assembly** from the shortcut menu.
  - c. Right-click the `ClientInteractServerEndpoint-SOA.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 39](#)
- [Task B, Configure the Service Unit, page 39](#)
- [Task C, Start Node and Install Shared Resources, page 40](#)
- [Task D, Deploy and Start the Service Assembly, page 40](#)
- [Task E, Run the Agent, page 40](#)

### 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 `SOA_CustomerRequest.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

This example adopts an ActiveMatrix Service JMS. You need to install the corresponding Shared Resource to the node.

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 Agent

In Lotus Notes:

1. Select the request document named `rpcClient` in the Request Response Invocation View.
2. From the Actions menu, select the HTTP agent that is created when setting up this example.
3. Click `6. Client Request Processing` under the TIB Administration folder. The status of the request should be `Processed`.

## Expected Results

---

Double-click the request document in the database `sample.nsf`, and the query result of the documents are displayed in the Return Data field.



## Chapter 6 **Publisher Binding with BWSE**

This example explains how to configure Publication Service in TIBCO Business Studio, wire it with BWSE service and run it in TIBCO ActiveMatrix Administrator.

### Topics

---

- [Example Description, page 44](#)
- [Setting Up the Example, page 45](#)
- [Running the Example, page 48](#)
- [Expected Results, page 50](#)

## Example Description

---

This example explains how to configure Publication Service in TIBCO Business Studio, wire it with BWSE service and run it in TIBCO ActiveMatrix Administrator.

## Setting Up the Example

---

To set up the example, complete the following tasks:

- [Task A, Set up Lotus Notes Application, page 45](#)
- [Task B, Create Agents, page 45](#)
- [Task C, Import the Adapter Project, Designer Project, and SOA Project, page 46](#)
- [Task D, Customize the Adapter Project, page 46](#)
- [Task E, Update the Designer Project, page 47](#)
- [Task F, Refresh Service Assembly and Build Archive, page 47](#)

### Task A Set up Lotus Notes Application

The `source.nsf` database is used in this example.

1. Start Lotus Notes and click **Databases**.
2. Open the `source.nsf` file from `ADLN_HOME/examples/LotusDatabase`, and upload them to Domino Server.
3. Configure the database.
  - a. Select the database file `source.nsf`.
  - b. Enter values for the following fields in the Setup Document:
    - `Queue Db Server`: IP address of the machine that hosts the database file.
    - `Host Name`: IP address of the machine on which the adapter is running.
    - `Port Number`: should be the same as the `Http Port to Listen to LotusNotes Events` field value that is specified in the Adapter Services tab. In this example, it is 8081.
  - c. Save the changes.

### Task B Create Agents

1. In Lotus Notes, select **Create > Agents**.
2. In the Name field, enter a name that indicates the action to be taken.  
For example, `HTTPStandardPublisher`.
3. Close the **Agent** dialog.
4. In the Action panel, select **Imported Java** from the Run list box.

5. Click **Import Class Files**. Import the class file `StandardPublisherHttp.class` from `ADLN_HOME/lib/agents/HTTP`.
6. Click **OK**.
7. Save the configuration.

### Task C Import the Adapter Project, Designer Project, and SOA Project

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\adlnse\6.0\samples\AMX_BWSE\PublisherBindBwse.zip`.
5. Ensure that the Adapter Project `PubWireWithBwse-UI`, the Designer Project `PubWireWithBwse-DABS`, and the SOA Project `PubWireWithBwse-SOA` have been selected in the Projects pane.
6. Select the **Copy projects into workspace** checkbox.
7. Click **Finish**.

### Task D Customize the Adapter Project

After importing the projects, you need to change the connection parameters based on the environment.

1. In the Project Explorer, expand `PubWireWithBwse-UI > defaultVars`, and then double-click `defaultVars.substvar` to open the editor for substitution variables.
2. Change values for substitution variables `adln.connection.dominoserver`, `adln.connection.username`, and `adln.connection.password`.
3. Save the configuration.
4. In the Project Explorer, expand `PubWireWithBwse-UI > Shared Resources`, and then double-click `LotusNotes Connection.shareInc` to open the editor for connection.
5. Click the **Test Connection** button to ensure that the adapter can connect to Lotus applications.



6. Regenerate WSDL file for the adapter.
  - a. In the Project Explorer, expand PubWireWithBwse-UI > Adapter Configurations, right-click the adapter instance ADLN\_PubBwseConfiguration.adlnmodel, and then select **Generate Adapter WSDL** from the shortcut menu.
  - b. In the Target Project dialog, select the SOA project PubWireWithBwse-SOA, click **OK**.
  - c. Click **OK** when prompted to overwrite the existing resource.

### Task E Update the Designer Project

1. In the Project Explorer, select the Designer Project PubWireWithBwse-DABS.
2. Switch to the Designer - TIBCO perspective. From the Designer menu, select **Project**, and then select **Import Resources from File, Folder, URL**.
3. In the Import Resource or File dialog, select **Folder** in the Format field, and then browse to select the schema folder in the SOA Project PubWireWithBwse-SOA. Click **OK**.
4. Import the ADLN\_PubBwseConfiguration\_genresources folder in the SOA Project PubWireWithBwse-SOA in the same way.
5. In the Project Explorer, expand PubWireWithBwse-SOA > Composites, and then right-click the composite file PubWireWithBwse-SOA.composite, select **Refresh** to update the BusinessWorks component.
6. Configure the process in the Designer Project.
  - a. In the Project Explorer, select the Designer Project PubWireWithBwse-DABS, and then double-Click the process file ADLN\_PubBwseConfiguration\_publisher.process.
  - b. In the Editor view, select the **Receive Partner Notification** activity, check the configuration.

### Task F Refresh Service Assembly and Build Archive

1. In the Project Explorer, expand PubWireWithBwse-SOA > Deployment Packages.
2. Right-click the service assembly file PubWireWithBwse-SOA.saf, and then select **Refresh Service Assembly** from the shortcut menu.
3. Right-click the PubWireWithBwse-SOA.saf file, and then select **Build Archive** from the shortcut menu.
4. 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 48](#)
- [Task B, Configure the Service Unit, page 48](#)
- [Task C, Start Node and Install Shared Resources, page 49](#)
- [Task D, Deploy and Start the Service Assembly, page 49](#)
- [Task E, Run the Agent, page 49](#)

### 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 `PubWireWithBwse.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 Agent

In Lotus Notes:

1. Select the Demo folder in the left hand pane.
2. Select the pre-defined publisher agent from the Actions menu. For example, select **HTTPStandardPublisher**.

## Expected Results

---

The selected documents in the Demo view are published successfully. The published documents with Appropriate symbols can be found in the Publishing Status view.

## Chapter 7

## RPC Server Binding with BWSE

This example explains how to configure RPC Server in TIBCO Business Studio, wire it with BWSE service and run it in TIBCO ActiveMatrix Administrator.

### Topics

---

- [Example Description, page 52](#)
- [Setting Up the Example, page 53](#)
- [Running the Example, page 56](#)
- [Expected Results, page 58](#)

## Example Description

---

This example demonstrates how to configure Request-Response Service to perform an insert operation with BWSE.

## Setting Up the Example

---

To set up the example, complete the following tasks:

- [Task A, Set up Lotus Notes Application, page 53](#)
- [Task B, Import the Adapter Project, Designer Project, and SOA Project, page 53](#)
- [Task C, Customize the Adapter Project, page 54](#)
- [Task D, Update the Designer Project, page 54](#)
- [Task E, Refresh Service Assembly and Build Archive, page 55](#)

### Task A Set up Lotus Notes Application

The `source.nsf` database is used in this example.

1. Start Lotus Notes and click **Databases**.
2. Open the `source.nsf` file from `ADLN_HOME/examples/LotusDatabase`, and upload them to Domino Server.
3. Select the `source.nsf` file that is at the Domino Server. Enter value for the following field in the Setup Document:
  - Queue Db Server: IP address of the machine that hosts the database file.
  - Host Name: IP address of the machine on which the adapter is running.
4. Save the changes.

### Task B Import the Adapter Project, Designer Project, and SOA Project

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\adlnse\6.0\samples\AMX_BWSE\RpcsBindBwse.zip`.
5. Ensure that the Adapter Project `RpcsWireWithBwse-UI`, the Designer Project `RpcsWireWithBwse-DABS`, and the SOA Project `RpcsWireWithBwse-SOA` have been selected in the Projects pane.

6. Select the **Copy projects into workspace** checkbox.
7. Click **Finish**.

### Task C Customize the Adapter Project

After importing the projects, you need to change the connection parameters based on the environment.

1. In the Project Explorer, expand `RpcsWireWithBwse-UI > defaultVars`, and then double-click `defaultVars.substvar` to open the editor for substitution variables.
2. Change values for substitution variables `adln.connection.dominoserver`, `adln.connection.username`, and `adln.connection.password`.
3. Save the configuration.
4. In the Project Explorer, expand `RpcsWireWithBwse-UI > Shared Resources`, and then double-click `LotusNotes Connection.shareInc` to open the editor for connection.
5. Click the **Test Connection** button to ensure that the adapter can connect to Lotus applications.
6. Regenerate WSDL file for the adapter.
  - a. In the Project Explorer, expand `RpcsWireWithBwse-UI > Adapter Configurations`, right-click the adapter instance `ADLN_RpcsBwseConfiguration.adlnmodel`, and then select **Generate Adapter WSDL** from the shortcut menu.
  - b. In the Target Project dialog, select the SOA project `RpcsWireWithBwse-SOA`, click **OK**.
  - c. Click **OK** when prompted to overwrite the existing resource.

### Task D Update the Designer Project

1. In the Project Explorer, select the Designer Project `RpcsWireWithBwse-DABS`.
2. Switch to the Designer - TIBCO perspective.
3. From the Designer menu, select **Project**, and then select **Import Resources from File, Folder, URL**.
4. In the Import Resource or File dialog, select **Folder** in the Format field, and then browse to select the schema folder in the SOA Project `RpcsWireWithBwse-SOA`. Click **OK**.
5. Import the `ADLN_RpcsBwseConfiguration_genresources` folder in the SOA Project `RpcsWireWithBwse-SOA` in the same way.



6. Configure the process in the Designer Project.
  - a. In the Project Explorer, select the Designer Project `RpcsWireWithBwse-DABS`, and then double-Click the process file `ADLN_RpcsBwseConfiguration_rpcServer.process`.
  - b. In the Editor view, select the **Timer** activity, the time interval is set to 10 seconds, which means the insert operation will be carried out every ten seconds. You can change the timer interval if necessary.
  - c. Select the **Invoke Partner** activity, check the data to be inserted.

### Task E Refresh Service Assembly and Build Archive

1. In the Project Explorer, expand `RpcsWireWithBwse-SOA > Deployment Packages`.
2. Right-click the service assembly file `RpcsWireWithBwse-SOA.saf`, and then select **Refresh Service Assembly** from the shortcut menu.
3. Right-click the `RpcsWireWithBwse-SOA.saf` file, and then select **Build Archive** from the shortcut menu.
4. 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 56](#)
- [Task B, Configure the Service Unit, page 56](#)
- [Task C, Start Node and Install Shared Resources, page 57](#)
- [Task D, Deploy and Start the Service Assembly, page 57](#)

### 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 `SOA_CustomerRequest.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.
3. Two service units are listed in the Service Units table, one is `JMS`, the other is `LotusNotesAdapter`. Do the following steps for both of them:
  - 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

This example adopts an ActiveMatrix Service JMS. You need to install the corresponding Shared Resource to the node.

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
```

## Expected Results

---

A new document is inserted in the view `Demo` of database `source.nsf`.

## Chapter 8

# Subscriber with Dynamic Login

This example shows how to use Subscription Service to perform insert operations with dynamic login.

## Topics

---

- [Example Description, page 60](#)
- [Setting Up the Example, page 61](#)
- [Running the Example, page 64](#)
- [Expected Results, page 66](#)

## Example Description

---

The example uses the Demo form included in the `source.nsf` file to perform Insert operation.

The example archive file can be found in the following location:

`AMX_HOME\extensions\adlnse\6.0\samples\AMX_BWSE`

## Setting Up the Example

---

To set up the example, complete the following tasks:

- [Task A, Set up Lotus Notes Application, page 61](#)
- [Task B, Import the Adapter Project, Designer Project, and SOA Project, page 61](#)
- [Task C, Customize the Adapter Project, page 62](#)
- [Task D, Update the Designer Project, page 62](#)
- [Task E, Refresh Service Assembly and Build Archive, page 63](#)

### Task A Set up Lotus Notes Application

The `source.nsf` database is used in this example.

1. Start Lotus Notes and click **Databases**.
2. Open the `source.nsf` file from `ADLN_HOME/examples/LotusDatabase`, and upload them to Domino Server.
3. Configure the database.
  - a. Select the database file `source.nsf`.
  - b. Enter value for the following field in the Setup Document:
    - Queue Db Server: IP address of the machine that hosts the database file.
    - Host Name: IP address of the machine on which the adapter is running.
  - c. Save the changes.

### Task B Import the Adapter Project, Designer Project, and SOA Project

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\adlnse\6.0\samples\AMX_BWSE\DynamicSubBin  
dBwse.zip`.

5. Ensure that the Adapter Project `DynamicSubWithBwse-UI`, the Designer Project `DynamicSubWithBwse-DABS`, and the SOA Project `DynamicSubWithBwse-SOA` have been selected in the Projects pane.
6. Select the **Copy projects into workspace** checkbox.
7. Click **Finish**.

### Task C Customize the Adapter Project

After importing the projects, you need to change the connection parameters based on the environment.

1. In the Project Explorer, expand `DynamicSubWithBwse-UI > defaultVars`, and then double-click `defaultVars.substvar` to open the editor for substitution variables.
2. Change values for substitution variables `adln.connection.dominoserver`, `adln.connection.username`, and `adln.connection.password`.
3. Save the configuration.
4. In the Project Explorer, expand `DynamicSubWithBwse-UI > Shared Resources`, and then double-click `LotusNotes Connection.shareInc` to open the editor for connection.
5. Click the **Test Connection** button to ensure that the adapter can connect to Lotus applications.
6. Regenerate WSDL file for the adapter.
  - a. In the Project Explorer, expand `DynamicSubWithBwse-UI > Adapter Configurations`, right-click the adapter instance `ADLN_DynamicSubBwseConfiguration.adlnmodel`, and then select **Generate Adapter WSDL** from the shortcut menu.
  - b. In the Target Project dialog, select the SOA project `DynamicSubWithBwse-SOA`, click **OK**.
  - c. Click **OK** when prompted to overwrite the existing resource.

### Task D Update the Designer Project

1. In the Project Explorer, select the Designer Project `DynamicSubWithBwse-DABS`.
2. Switch to the Designer - TIBCO perspective.
3. From the Designer menu, select **Project**, and then select **Import Resources from File, Folder, URL**.



4. In the Import Resource or File dialog, select **Folder** in the Format field, and then browse to select the `schema` folder in the SOA Project `DynamicSubWithBwse-SOA`. Click **OK**.
5. Import the `ADLN_DynamicSubBwseConfiguration_genresources` folder in the SOA Project `DynamicSubWithBwse-SOA` in the same way.
6. In the Project Explorer, expand `DynamicSubWithBwse-SOA > Composites`, and then right-click the composite file `DynamicSubWithBwse-SOA.composite`, select **Refresh** to update the BusinessWorks component.
7. Configure the process in the Designer Project.
  - a. In the Project Explorer, select the Designer Project `DynamicSubWithBwse-DABS`, and then double-click the process file `ADLN_DynamicSubBwseConfiguration_subscriber.process`.
  - b. In the Editor view, select the **Timer** activity, the time interval is set to 10 seconds, which means the insert operation will be carried out every ten seconds. You can change the time interval if necessary.
  - c. Select the **LoginAction** activity, in the Input tab, change the values of the username field and the password field to login to Domino Server.
  - d. Select the **SubscriberInsert** activity, check the configuration of the insert operation in the Input tab.

### Task E Refresh Service Assembly and Build Archive

1. In the Project Explorer, expand `DynamicSubWithBwse-SOA > Deployment Packages`.
2. Right-click the service assembly file `DynamicSubWithBwse-SOA.saf`, and then select **Refresh Service Assembly** from the shortcut menu.
3. Right-click the `DynamicSubWithBwse-SOA.saf` file, and then select **Build Archive** from the shortcut menu.
4. 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 64](#)
- [Task B, Configure the Service Unit, page 64](#)
- [Task C, Start Node and Install Shared Resources, page 65](#)
- [Task D, Deploy and Start the Service Assembly, page 65](#)

### 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 `DynamicSubWithBwse-SOA.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
```

## Expected Results

---

- Once the insert operation has been performed, the TIBCO ActiveMatrix BusinessWorks process logs out the user who performed the operation.
- Open Lotus Notes and check the view `Demo` in `source.nsf`. New documents are inserted.

## Chapter 9      **EAR to SA Example**

This example explains how to convert an EAR file to service assembly in TIBCO Business Studio and then run it in TIBCO ActiveMatrix Administrator.

### Topics

---

- [Example Description, page 68](#)
- [Setting Up the Example, page 69](#)
- [Running the Example, page 71](#)
- [Expected Results, page 73](#)

## Example Description

---

This example demonstrates how to convert an EAR file to service assembly in TIBCO Business Studio and then run it in TIBCO ActiveMatrix Administrator.

The example archive file can be found in the following location:

`AMX_HOME\extensions\adlnse\6.0\samples\AMX_EAR`

## Setting Up the Example

---

To set up the example, complete the following tasks:

- [Task A, Set up Lotus Notes Application, page 69](#)
- [Task B, Create Agents, page 69](#)
- [Task C, Create EAR File in TIBCO Designer, page 69](#)
- [Task D, Convert the EAR File to Service Assembly, page 70](#)

### Task A Set up Lotus Notes Application

The source .nsf database is used in this example.

1. Start Lotus Notes and click **Databases**.
2. Open the source .nsf file from `ADLN_HOME/examples/LotusDatabase`, and upload them to Domino Server.
3. Select the source .nsf file that is at the Domino Server. Enter value for the following field in the Setup Document:
  - Queue Db Server: IP address of the machine that hosts the database file.
  - Host Name: IP address of the machine on which the adapter is running.
4. Save the changes.

### Task B Create Agents

1. In Lotus Notes, select **Create > Agents**.
2. In the Name field, enter a name that indicates the action to be taken. For example, `HTTPStandardPublisher`.
3. In the action panel, select **Imported Java** from the Run list box.

Click **Import Class Files** to import the class file `HTTPStandardPublisherAgent` from `ADLN_HOME/lib/agents/HTTP`.
4. Click **OK** and save the configuration.

### Task C Create EAR File in TIBCO Designer

1. Unzip the example archive file.
2. Start TIBCO Designer and select **Open existing project**.

3. In the Multi-File Project tab, select PubEARDemo-Designer for the Project Directory field. Click **OK**.
4. Change the configuration as necessary. For example, provide values for the global variables `adln.connection.dominoserver`, `adln.connection.username`, and `adln.connection.password`.
5. Select the adapter instance named LotusNotesAdapterConfiguration, click **Test Connection** in the Design-time Connection tab to ensure that the adapter can connect to the Lotus application.
6. Select the PublicationService of the adapter instance, click **Get Schema** in the Configuration tab.
7. Select the Enterprise Archive PubEARDemo-Designer, change the field of File Location in the Configuration tab if necessary.
8. Click **Build Archive** in the Configuration tab to create the archive file.

#### Task D Convert the EAR File to Service Assembly

1. Start TIBCO Business Studio and create a new ActiveMatrix SOA project.



Do not use the SOA project named EARTransferDemo-SOA in the sample directory, which is only for reference.

2. From the File menu, select **Import**.
3. In the Import dialog, select **General > File System**. Click **Next**.
4. Check the **Select root directory** radio button, browse to select the directory that maintains the EAR file PubEARDemo-Designer.ear. Specify the destination in the Into folder field.
5. Click **Finish**.
6. In the Project Explorer, right-click the imported file PubEARDemo-Designer.ear, and then select **Build Service Assembly Archive** from the shortcut menu.



## Running the Example

---

To run the example, complete the following tasks:

- [Task A, Upload Service Assembly Archive, page 71](#)
- [Task B, Configure the Service Unit, page 71](#)
- [Task C, Start Node and Install Shared Resources, page 72](#)
- [Task D, Deploy and Start the Service Assembly, page 72](#)
- [Task E, Run the Agent, page 72](#)

### 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 `PubEARDemo_Designer.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

This example adopts an ActiveMatrix Service JMS. You need to install the corresponding Shared Resource to the node.

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 Agent

In Lotus Notes:

1. Select the `Demo` folder in the left hand pane.
2. Select the pre-defined publisher agent from the Actions menu to trigger the `Demo` document. For example, select **HTTPStandardPublisher**.

## Expected Results

---

A new Demo document is published. The published document with Appropriate symbols can be found in the Publishing Status view.



# Index

## B

BWSE

    Publisher [43, 59](#)

    RPC Server [51](#)

## C

create node [5](#)

customer support [x](#)

## D

dynamic login

    RPC Server [25](#)

    Subscriber [59](#)

## E

environment [4](#)

## H

HTTPStandardPublisherAgent [35, 45](#)

## I

Import Class Files [18, 35, 46, 69](#)

## L

Lotus Notes database (nsf files) [3](#)

## N

Node [5](#)

## O

overview [2](#)

## P

prerequisites [3](#)

Publisher Binding with BWSE [43, 59](#)

## R

RPC Server Binding with BWSE [51](#)

## S

Service Unit [12](#)

SOAP [10](#)

    RPC Client [15](#)

    RPC Server [25](#)

    RPC Server and RPC Client [33](#)

    Subscriber [7](#)

Subscriber Binding with SOAP [7](#)  
support, contacting [x](#)

## T

technical support [x](#)  
TIBCO\_HOME [viii](#)

## U

upload Service Assembly archive [12, 21](#)