TIBCO ActiveMatrix® Adapter for Kenan/BP

Examples

Software Release 6.0
November 2009

Important Information

Contents

Preface

TIBCO ActiveMatrix Adapter for Kenan/BP ships with pre configured examples. This manual describes how to run these examples, and explains the configuration of each.

Topics

Related Documentation

This section lists documentation resources you may find useful.

TIBCO ActiveMatrix Adapter for Kenan/BP Documentation

The following documents form the TIBCO ActiveMatrix Adapter for Kenan/BP documentation set:

- *TIBCO ActiveMatrix Adapter for Kenan/BP Concepts* Read this manual to gain an understanding of the product that you can apply to the various tasks you may undertake.
- *TIBCO ActiveMatrix Adapter for Kenan/BP Installation* Read this manual to learn how to install TIBCO ActiveMatrix Adapter for Kenan/BP.
- *TIBCO ActiveMatrix Adapter for Kenan/BP Configuration and Deployment* Read this manual for instructions on creating, configuring , and deploying adapter projects.
- *TIBCO ActiveMatrix Adapter for Kenan/BP Examples* Read this manual to work through the examples provided with the adapter.
- *TIBCO ActiveMatrix Adapter for Kenan/BP Release Notes* Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for this release.

Other TIBCO Product Documentation

You may find it useful to read the documentation for the following TIBCO products. Note that only books that relate to adapters are listed. Each of the books is available from the `doc` directory in the product's installation area.

- TIBCO Designer™
- TIBCO Administrator™
- TIBCO ActiveMatrix BusinessWorks™
- TIBCO Rendezvous®
- TIBCO Enterprise Message Service™
- TIBCO Hawk®
- TIBCO Adapter™ SDK
- TIBCO Runtime Agent™

Third-Party Documentation

You may also find it useful to read the followingKenan/BP documentation from:

- *API TS Guide*
- *API TS Reference*

Typographical Conventions

The following typographical conventions are used in this manual.

Table 1 General Typographical Conventions

Convention	Use
TIBCO_HOME *ENV_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`. 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.
`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]`
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*

Table 1 General Typographical Conventions (Cont?)

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`

Table 2 Syntax Typographical Conventions

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 includes procedures and prerequisites that have to be carried out before you start running the examples provided with your TIBCO ActiveMatrix Adapter for Kenan/BP installation. Detailed information for the examples is provided in subsequent chapters of this guide.

The examples work properly with Kenan 12.0 XSD Schemas. For other versions such as 11.x, users have to import the corresponding set of XSD Schemas, and recreate the TIBCO ActiveMatrix BusinessWorks rendering activities to render the XML request correctly.

Topics

Overview

The examples discussed in this guide demonstrate:

- How to configure the TIBCO ActiveMatrix Adapter for Kenan/BP and a Request Response Service for the same.
- How to configureTIBCO ActiveMatrix BusinessWorksfor integrating with the Request-Response Service.
- How to make a Kenan/BP API Call.
- How to call a custom function using the Request-Response Service.
- How to use the Ordered set of API calls.

The Business Objects used in the examples are:

- Account
- Product
- Order
- Service
- OrderedAccount

For more information on how to integrate the adapter with TIBCO ActiveMatrix BusinessWorks, refer to *TIBCO ActiveMatrix Adapter for Kenan/BP Configuration and Deployment*.

The examples provide different operations on each of these Business Objects. A brief of these operations is given below:

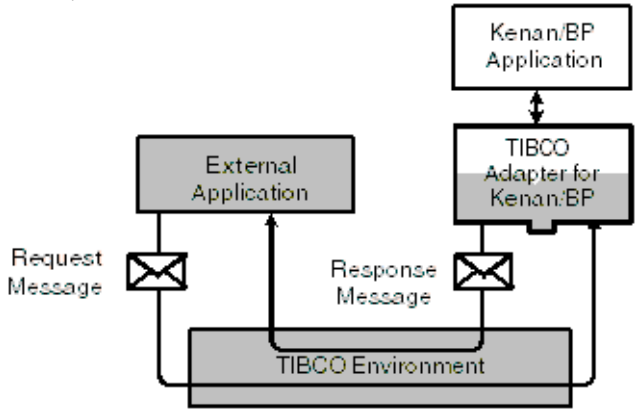
Get

In this operation, an external application (for example, TIBCO ActiveMatrix BusinessWorks) requests data to be fetched from the Kenan/BP application. The adapter Request-Response Service extracts the search criteria from the incoming message (RV/JMS) sent by the external application. After doing so, the process writes the output of the search operation to a file.

Create

In this operation, an external application (for example, TIBCO ActiveMatrix BusinessWorks) sends data to the adapter Request-Response Service. This in turn sends a create request to the Kenan/BP application, which creates the Business Entity provided the input data satisfies the Kenan/BP data validation criteria.

Figure 1 Request-Response Service



Using TIBCO ActiveMatrix BusinessWorks

If you are using the adapter with TIBCO ActiveMatrix BusinessWorks, the following software must be installed to run the examples:

- TIBCO ActiveMatrix BusinessWorks
- TIBCO Administrator
- TIBCO ActiveMatrix Adapter for Kenan/BP
- Kenan/BP 11.5, 11.7 or 12.0
- TIBCO Runtime Agent

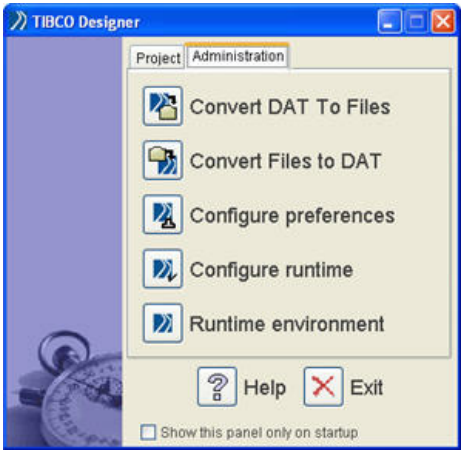
TIBCO Enterprise Message Service must be installed to run examples that use JMS as the transport. The JMS server must be running and accessible to the UNIX machine on which the adapter is installed. Kenan/BP application must be installed and running on the UNIX machine where the adapter is installed.

In these examples, a TIBCO ActiveMatrix BusinessWorks process sends a request to the adapter Request-Response Service. An XML message containing the details of the API Call to be executed is embedded in the incoming request. Based on this message, the adapter executes the request and sends the output back to the calling process.

Working with DAT Files

You cannot directly open a dat file in TIBCO Designer and make modifications to the configurations. To do this, carry out the following steps:

1. Convert the dat file to a multi-file project.
   a. Open TIBCO Designer. In the first screen that is displayed, click **Administration**.



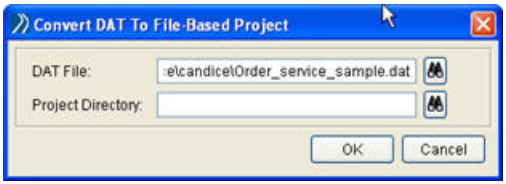
   b. Then click the **Convert DAT to Files** icon. In the window that is displayed, browse the DAT File field and select the dat file you wish to convert to a multi-file project. Thereafter, browse the Project Directory field and select the location (other than the folder containing the dat file) where you wish to save the multi-file project and give the same name as your dat file. Click **OK**.

If you use a different name for the multi-file project, the value of the global variable deployment will be changed. Therefore, it is mandatory to use the same name for the multi-file project as that of the .dat file.

In the **Convert DAT To File-Based Project** dialog, you can see the `CREATE.dat` file in the `temp` folder. Do not choose `temp` to save your

multi-file project, as TIBCO Designer does not allow you to save a multi-file project with the same name as that of the .dat file in the same folder.

You can save the multi-file project in any other folder, with the same name as that of the dat file.



2. Click the **Open existing project** icon from the first TIBCO Designer screen. Browse to the directory where the converted multi-file project is saved.
3. Make configuration changes per your requirements.
4. Export the multi-file project to a dat. Select **Project>Export Full Project** from the menu. Browse and select the location of the directory you wish to save the dat file to. Ensure that the directory is different from the multi-file project. Enter the name of the project and click **OK**.

Examples Location

The files for the examples along with the sample messages are in the following location:

For TIBCO ActiveMatrix BusinessWorks examples:

`$TIBCO_HOME\adapter\adkenan\<version_num>\examples\BusinessWorks`

This folder contains two examples:
`Account_Get_RV.zip` and `Product_Create_JMS.zip`.

In addition to this, it also contains an example titled `OrderedAccount_Create_RV.zip`. This demonstrates the adapter's interaction with the Ordered set of API Calls.

For Custom function callout examples:

`$TIBCO_HOME\adapter\adkenan\<version_num>\examples\BusinessWorks\K`enanCustomCallOut

This folder contains two subfolders `CustomClasses` (containing sample Java code for writing custom classes) and `Repository` (containing a TIBCO ActiveMatrix BusinessWorks project to demonstrate the same). It also contains a sample XSD (XML Schema Definition) which is used by the TIBCO ActiveMatrix BusinessWorks project to generate an XML message to be processed by the adapter.

Chapter 2 TIBCO ActiveMatrix BusinessWorks: Working with the AccountGet API Call

This chapter demonstrates the execution of the AccountGet API Call using TIBCO ActiveMatrix BusinessWorks.

Topics

Example Description

This example demonstrates the usage of the `AccountGet` API Call. The incoming request to the adapter consists of an XML message which has the `AccountGet` node. This XML block has an Account ID using which the application retrieves the Account details. The response from Kenan/BP is also in the form of an XML message which contains the `AccountGetResponse` node.

Configuring the Example

Task A Configure an adapter instance

5. Start TIBCO Designer.
   Click **Programs > TIBCO > TIBCO Designer <version_num> > Designer <version_num>**.
6. In the initial dialog box, click **New Empty Project** and specify a name for the project in the dialog box titled **Save Project**. Click **OK**.
7. Drag and drop an adapter instance from the palette panel to the design panel.
8. Provide a name for the adapter instance in the Instance Name field in the **Configuration** tab.
9. Select the **Global Variables** tab in the project panel and provide the value for the following global variables:
   - `adkenanLogin` (The User name to connect to the Security Server)
   - `adkenanPassword` (The password to connect to the Security Server)
   - `adkenanSecurityRealm` (Security Realm variable for the Security Server)
10. Save the project by selecting **Project > Save**.
11. Exit TIBCO Designer.

Task B Convert the project into a DAT file

12. Start TIBCO Designer.
    Click **Programs > TIBCO > TIBCO Designer <version_num>> Designer <version_num>**.
13. In the first screen that is displayed, click the **Administration** tab.
14. Then click the **Convert files to DAT** icon.
15. Browse the Project Directory field and select the location where the project had been saved (point 6 in task A: Configure an adapter instance).
16. Browse the .dat file field and select the location where you intend to save the .dat file. The only restriction here is that there cannot be a .dat file as well as a project of the same name in the same directory.
17. Click **OK** to generate the .dat file in the specified location.

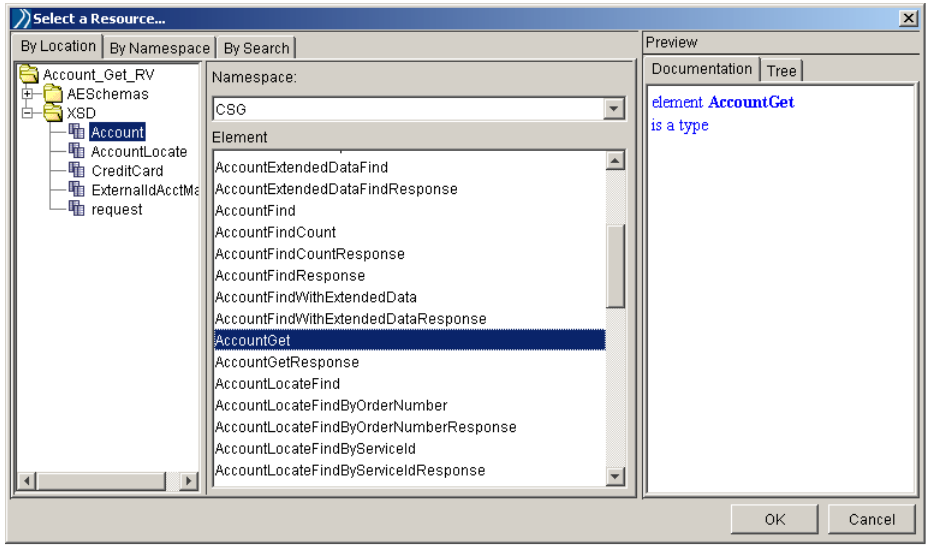
Task C Importing the XSD files

18. Start TIBCO Designer.
    Click **Programs > TIBCO > TIBCO Designer <version_num> > Designer <version_num>**.
19. Open the project saved in Task A. This can be done by clicking **Open Existing Project** and selecting the project.
20. Create a folder named `XSD` under the root folder. This will store all the imported XSD files. Create another folder called `Process` to store the BusinessWorks process.
21. For Kenan/BP 11.5 or 11.7, import the following XSD files in the order specified: `request.xsd, AccountLocate.xsd, ExternalIdAcctMap.xsd, CreditCard.xsd and Account.xsd.` For Kenan/BP 12.0, import the following XSD files in the order specified: `request.xsd, AccountLocate.xsd, ExternalIdAcctMap.xsd, PaymentProfile.xsd and Account.xsd`. The order is useful because `Account.xsd` includes the schema location of the other XSD files. These XSD files are a part of the Kenan/FX installation.
22. To import an XSD, select **Project > Import Resources from File, Folder, URL**. Select the file (`.xsd, .xslt, .wsdl`) from the drop-down box and browse to point to the XSD location.

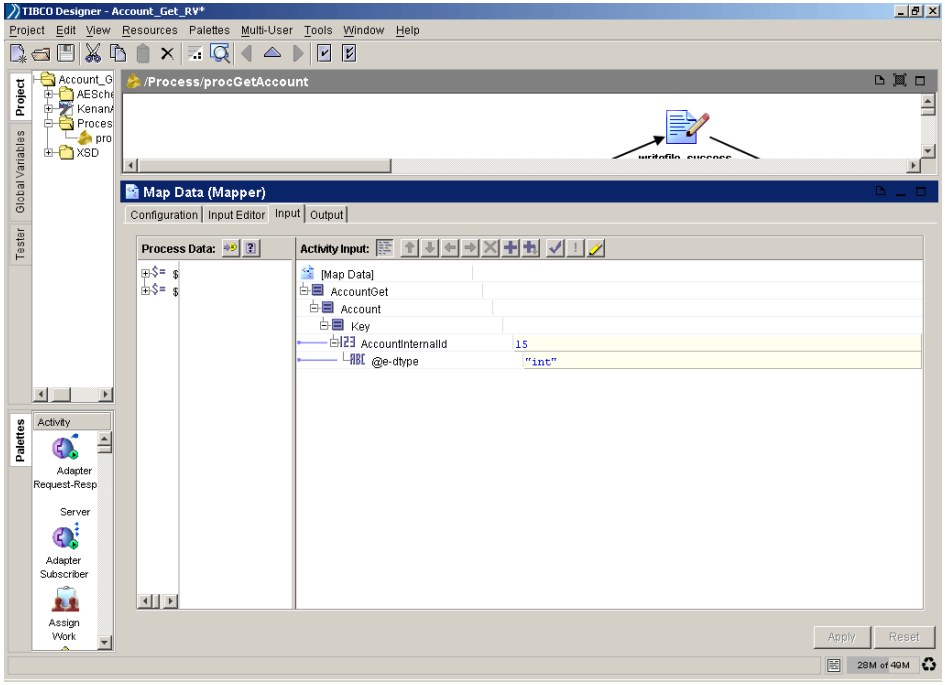
Task D Configuring the BusinessWorks process

23. Create a new TIBCO ActiveMatrix BusinessWorks process by dragging the **Process Definition** icon from the palette panel and dropping it in the design panel.

24. Configure a mapper task:
    - Drag and drop a mapper task from the palette panel.
    - Click the **Input Editor** tab.
    - Click the **Add Child** button and select **XML Element Reference** from the **Content** drop down box.
    - In the Schema field (which shows a default of `<No Namespace>`), browse and select the `AccountGet` element in `Account.xsd`.



    - Click **OK** in the dialog box shown above. Click **Apply** in the **Mapper task** dialog box.
    - Select the **Input** tab to provide the field values to execute this request.
    - Expand the `AccountGet` node in the Activity Input panel. Specify a value for the `AccountInternalId` and specify **`int`** in the e-dtype field.



— Click **Apply**.

— Attach a trigger from the Start node to the Mapper task.

25. Configure a Render-XML task

— Drag and drop a Render-XML task from the palette panel to the design panel.

— Attach a trigger from the Mapper task to the Render XML task.

— In the **Configuration** tab, select Output Style as text. Select the **Validate Input** and **Format with Default Namespace Prefix** check boxes. Select UTF-8 as the encoding.

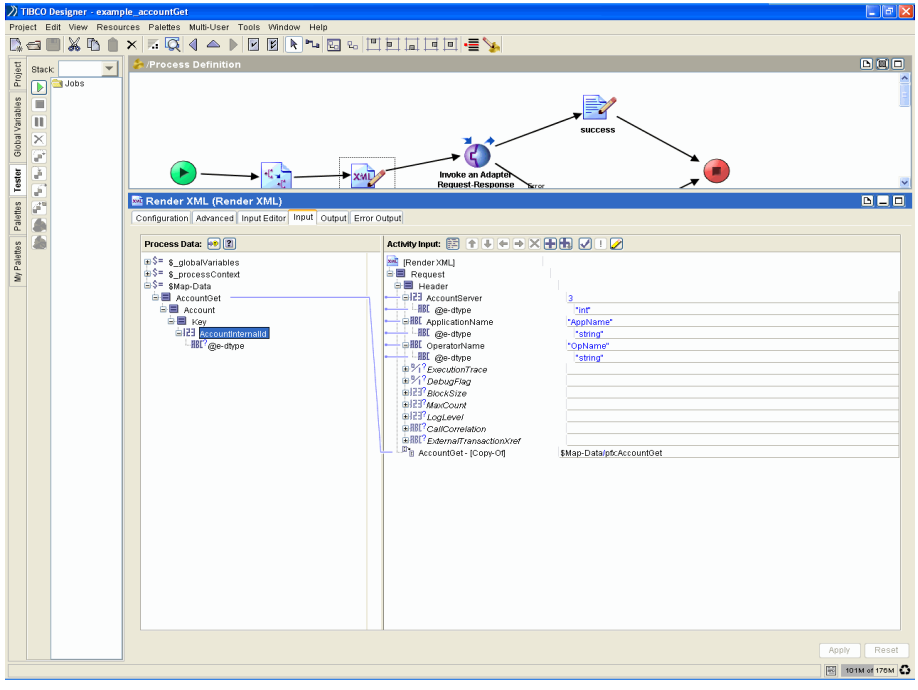
— In the **Input Editor** tab, select the `Request` element from `Account.xsd`. The procedure is exactly same as that of the Mapper task. (Refer to points c and d in the previous section).

— Select the **Input** tab to provide the field values to execute this request.

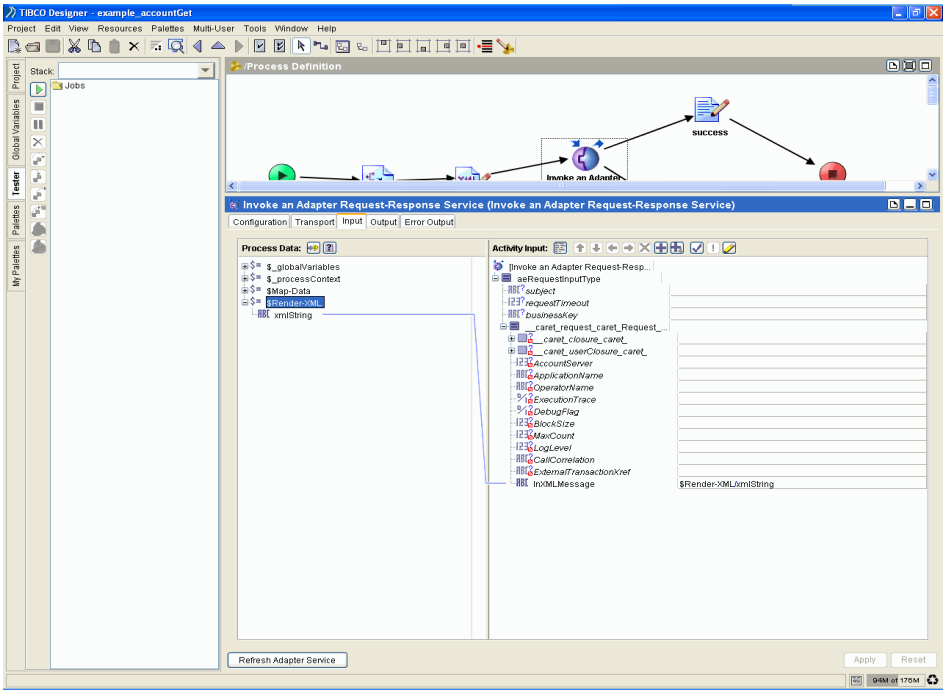
— Expand the `Request` node in the Activity Input panel. Specify values for the fields under the Header group.

— Drag and point the `AccountGet` node from the Process Data panel (this is the output of the mapper task) to the node named any element or exception.

— In the ensuing dialog box titled Mapping Wizard, select (any element). Click **Next** and then click **Finish**. The screen is shown next.



26. Configure an Invoke an Adapter Request-Response Service task
    - Drag and drop a Invoke an Adapter Request-Response Service task from the palette panel to the design panel.
    - In the Adapter Service field browse and select the Request-Response Service. Click **Apply**.
    - Attach a trigger from the Render XML task to the Invoke task.
    - Map the Render XML output (a string called `XMLstring`) to the incoming Request-Response Schemas (a string named `InXMLMsg`) as shown.



    - Click **Apply**.

The subsequent steps (like writing the response to a file) are optional.

Set up the AccountGet Example

Task E Importing the project

Before starting the example you must import the sample project zip file and save it in a new project.

1. Start TIBCO Designer.
   Click Programs > TIBCO > TIBCO Designer <version_num> > Designer<version_num>.
2. In the initial dialog box, click New Empty Project and specify a name for the project in the dialog box titled Save Project. Click **OK**.
3. Import the project by clicking **Project > Import Full Project**.
4. Select the **ZIP Archive** tab in the Import dialog box.
5. Browse to select the `Account_Get_RV.zip` file. By default, this is located in the `$TIBCO_HOME/adapter/adkenan/<version_num>/examples/BusinessWorks` folder. Click **OK**.
6. In the I**mport Options** dialog, select **Replace existing global variables with those in import** and **Overwrite on Name Conflict**. Click **Apply**.
7. For Kenan/BP 11.5 or 11.7, import the following XSD files in the order specified: `request.xsd, AccountLocate.xsd, ExternalIdAcctMap.xsd, CreditCard.xsd` and `Account.xsd`. For Kenan/BP 12.0, import the following XSD files in the order specified: `request.xsd, AccountLocate.xsd, ExternalIdAcctMap.xsd, PaymentProfile.xsd` and `Account.xsd`. The order is useful because, `Account.xsd` includes the schema location of the other XSD files. These XSD files are a part of the Kenan/FX installation.
8. Modify the global variables pertaining to the Security Server Settings as well as the `RvDaemon` parameter as required.
9. Select the **Header Fields** tab under the Request-Response Service. Enable the message header fields by selecting the **Specify Header Fields** check box. Specify the header fields per your settings. These header fields will be prefixed to the XML message sent to the adapter.
10. Save the project by selecting **Project > Save**.

Task F Converting the repository into .dat format

11. In TIBCO Designer, click **Project > Export Full Project**.

12. In the following dialog box, select the **Local Repository** tab.
13. Specify a name in the Project Name field and a valid path in the Dir Name. Click **Yes**.
14. In the **Create Project** dialog, click **OK**.

Running the Example

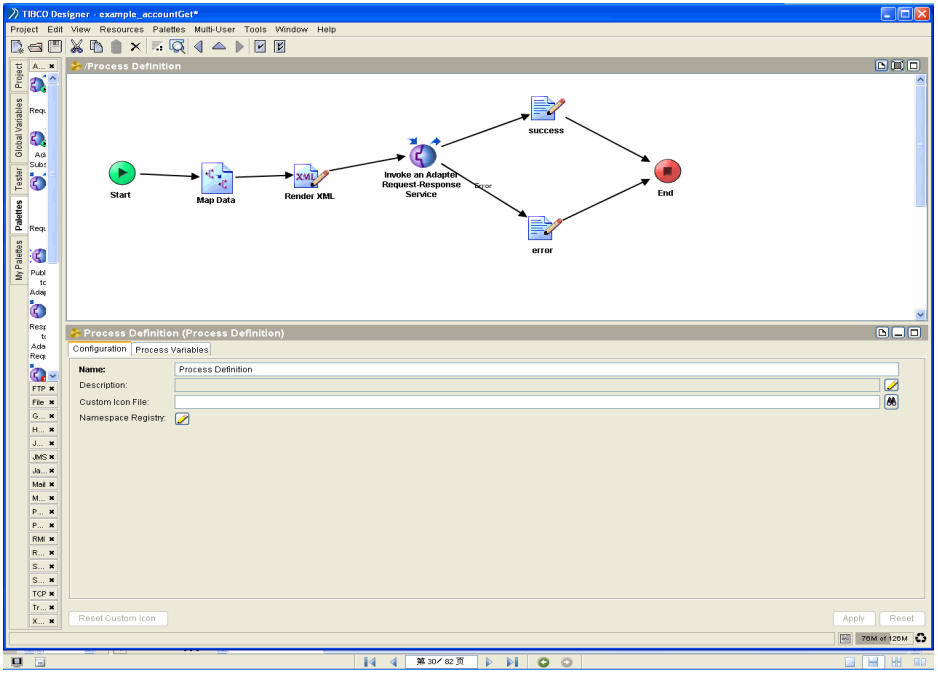
Perform the following tasks to run the example.

Task G Starting the Adapter

1. Edit the `adkenan.tra` file. This is located in `$TIBCO_HOME/adapter/adkenan/<version_num>/bin`. Specify the `repourl` as the location where the local repository was saved in the previous section (Refer to Task F, Converting the repository into .dat format). Specify the `configurl` as `/tibco/private/adapter/KenanAdapterConfiguration`.
2. Open a new console session on the UNIX machine where the adapter is installed and the Kenan/FX middleware is running.
3. Move to `$TIBCO_HOME/adapter/adkenan/<version_num>/bin` and type `adkenan`. This brings up the adapter and displays the message `Successfully initialized the adapter`.

Task H Running the TIBCO ActiveMatrix BusinessWorks Process

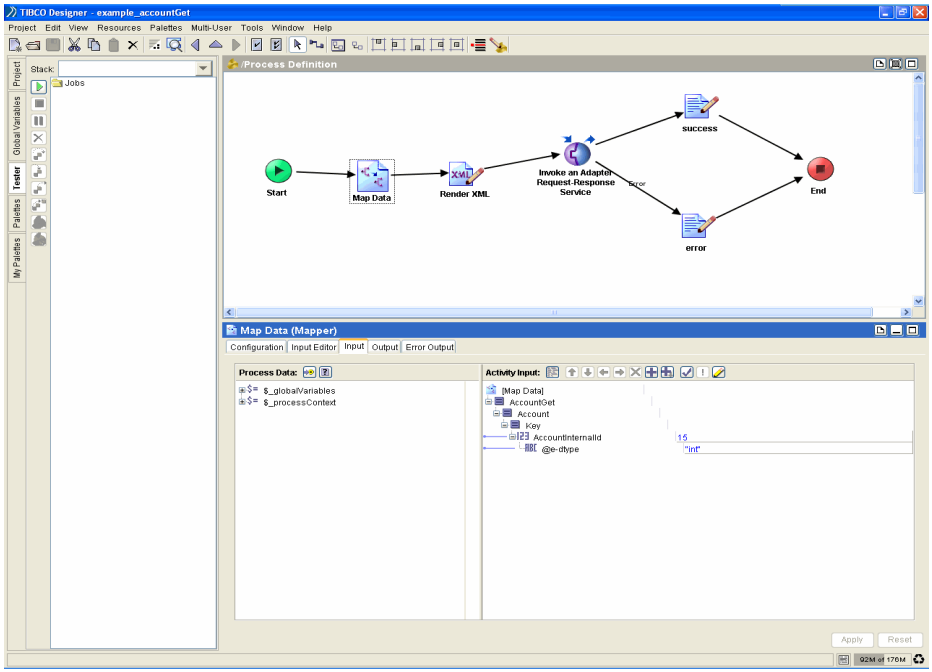
1. Start TIBCO Designer.
   Click **Programs > TIBCO > TIBCO Designer <version_num> > Designer <version_num>**.
2. In the initial dialog box, click **Open Existing Project**. Browse and select the location where the project was saved in Task E, Importing the project in the section, Set up the AccountGet Example.
3. Click on `procGetAccount` in the `Process` folder. The BusinessWorks process is displayed as shown next.



4. Check the value or the `RvDaemon` global variable to see whether it reflects the same Daemon parameter of the repository which is being used by the adapter to run.

5. Provide a valid location for the output file in both the Write File activities titled **success** and **error**. The file associated with success will have the reply XML message sent by Kenan/BP to the request sent by the adapter and the other one (associated with error) will have the Exception message being sent by the adapter, in case of errors. To provide a valid location for the output file, double-click the corresponding **Write File** icon, select the **Input** tab, and edit the filename parameter.

6. Provide a valid value for the `AccountInternalId` field in the Render-XML task. Double-click the **Render-XML** icon, select the **Input** tab, expand the `AccountGet` node till you see the `AccountInternalId` field. The adapter will search for the account with the same Internal ID. The Render-XML task will be displayed as shown next:

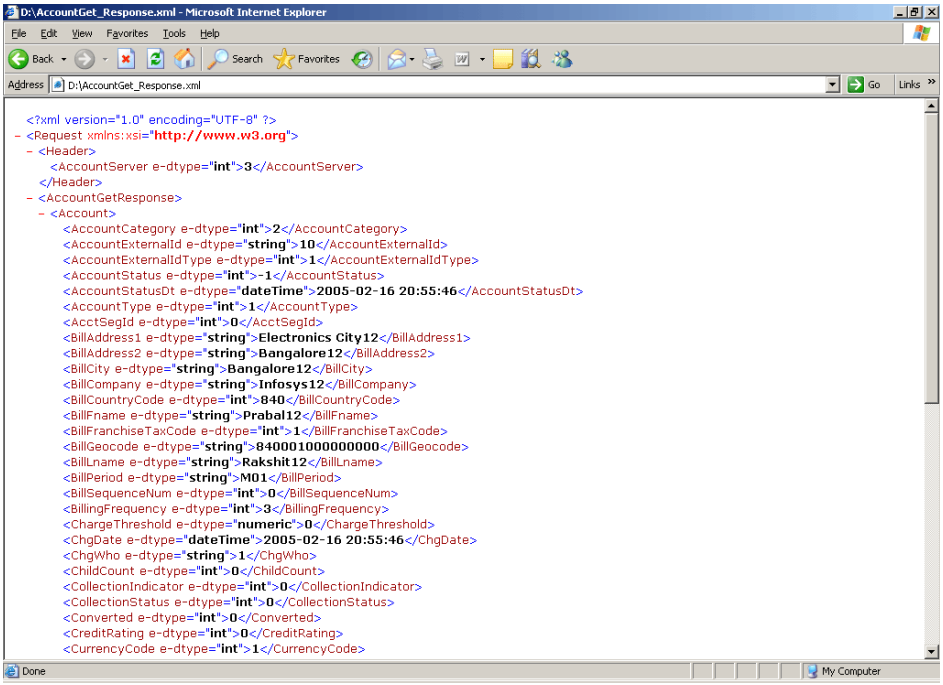


There are two Write File activities. The first Write File is used to store the result of successful API execution. The second Write File is used to store an exception, if any.

7. Trigger the TIBCO ActiveMatrix BusinessWorksprocess. Select the **Tester** tab in the project panel and press **F9**. In the next dialog box, click **Load & Start Current**.

Expected Results

- Check the output XML file (the path of which was provided in step 5). A sample output is shown next.



- The message `Successfully processed the request` is displayed on the adapter console.

Chapter 3

TIBCO ActiveMatrix BusinessWorks: Working with the ProductCreate API Call (Account Level)

This chapter demonstrates the execution of the ProductCreate API Call using TIBCO ActiveMatrix BusinessWorks.

Topics

Example Description

This example demonstrates the usage of the ProductCreate API Call. The incoming request to the adapter consists of an XML message which has the ProductCreate node. This XML block has details based on which Kenan/BP creates a Product. The response from Kenan/BP is also in the form of an XML message which contains the `ProductCreate` Response node.

Refer Configuring the Example on page 11 for details on how to configure the example.

Set up the ProductCreate Example

Task I Importing the project

Before starting the example, you must import the sample project zip file and save it in a new project.

1. Start TIBCO Designer.
   Click Programs > TIBCO > TIBCO Designer <version_num> > Designer <version_num>.
2. In the initial dialog box, click **New Empty Project** and specify a name for the project in the **Save Projec**t dialog. Click **OK**.
3. Import the project by clicking **Project > Import Full Project**.
4. Select the **ZIP Archive** tab in the Import dialog box.
5. Browse to select the `Product_Create_JMS.zip` file. By default, this is located in the `$TIBCO_HOME/adapter/adkenan/<version_num>/examples/BusinessWorks` folder. Click **OK**.
6. In the I**mport Options** dialog box, select **Replace existing global variables with those in import** and **Overwrite on Name Conflict**. Click **Apply**.
7. Import the following XSD files in the order specified: `Request.xsd`, `Product.xsd`. The order is useful because `Product.xsd` includes the schema location of the request XSD file. These XSD files are a part of the Kenan/FX installation.
8. Modify the global variables pertaining to the Security Server Settings as well as the `JmsProviderURL` parameter as required.
9. Select the **Header Fields** tab under the Request-Response Service. Enable the message header fields by selecting the **Specify Header Fields** check box. Specify the header fields per your settings. These header fields will be prefixed to the XML message sent to the adapter.
10. Save the project by selecting **Project > Save.**

Task J Converting the repository into .dat format

11. In TIBCO Designer, click **Project > Export Full Project**.
12. In the following dialog box, select the **Local Repository** tab.
13. Specify a name in the Project Name field and a valid path in the Dir Name. Click **Yes**.

14. In the following dialog box titled Create Project, click **OK**.

Running the Example

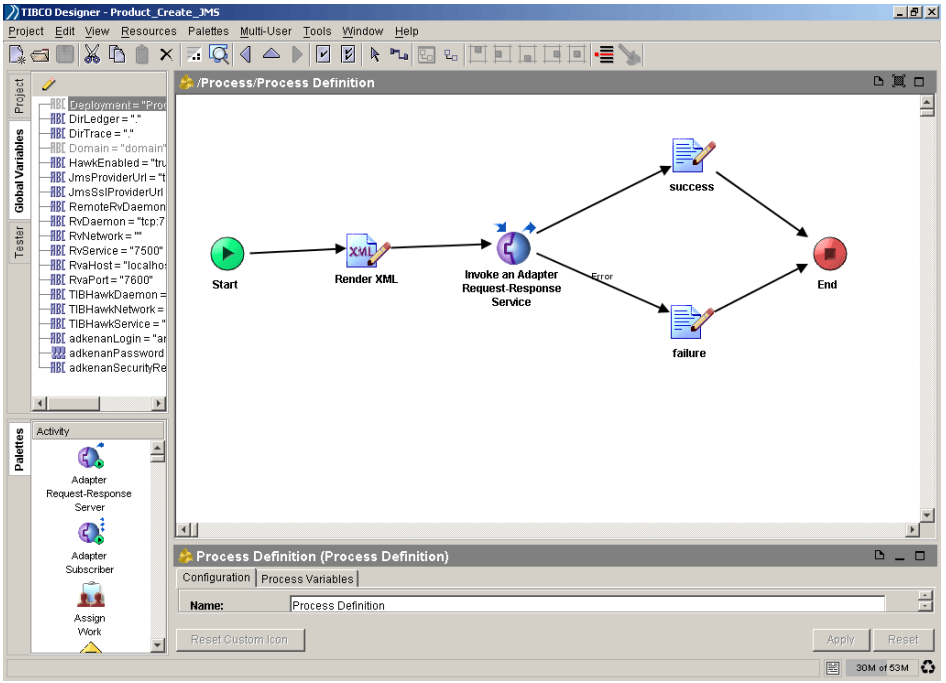
Perform the following tasks to run the example.

Task A Starting the Adapter

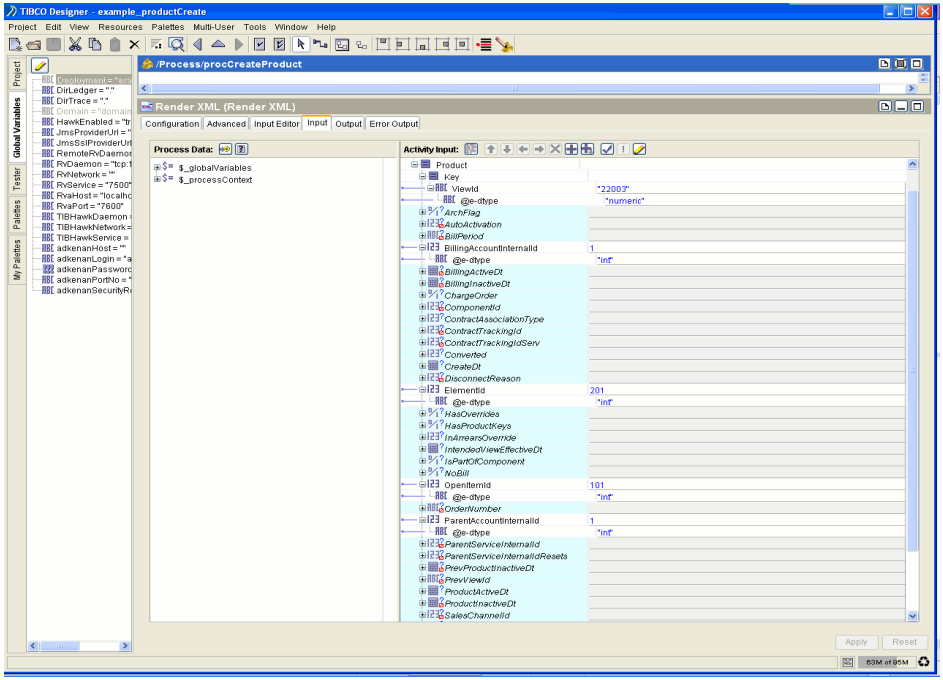
1. Edit the `adkenan.tra` file. This is located in `$TIBCO_HOME/adapter/adkenan/<version_num>/bin`. Specify the `repourl` as the location where the local repository was saved in the previous section. Specify the `configurl` as `/tibco/private/adapter/KenanAdapterConfiguration`.
2. Open a new console session on the UNIX machine where the adapter is installed and the Kenan/FX middleware is running.
3. Move to `$TIBCO_HOME/adapter/adkenan/<version_num>/bin` and type `adkenan`. This brings up the adapter and displays the message `Successfully initialized the adapter`.

Task B Running the TIBCO ActiveMatrix BusinessWorksProcess

1. Start TIBCO Designer.
   Click **Programs > TIBCO > TIBCO Designer <version_num> > Designer <version_num>**.
2. In the initial dialog box, click **Open Existing Project**. Browse and select the location where the project was saved.
3. Click `procCreateAccount` in the `Process` folder. The BusinessWorks process is displayed as shown next:



4. Check the value or the `JmsProviderUrl` global variable to see whether it reflects the same parameter of the repository which is being used by the adapter to run.
5. Provide a valid location for the output file in the Write File activity. This file contains the reply XML message sent by Kenan/BP to the request sent by the adapter. Double-click the **Write File** icon, select the **Input** tab and edit the filename parameter.
6. Provide a valid value for the input parameters in the Render-XML task. Double-click the **Render-XML** icon, select the **Input** tab and provide the values as desired. The datatype of a particular input field has to be mentioned against the 'e-dtype' tag for it. The Render-XML task will be displayed as shown next:



In this example, the following parameters gave been populated:

— `BillingAccountInternalId`

— `ElementId (Specific for Account level Products).`

— `OpenItemId`

— `ParentAccountInternalId`

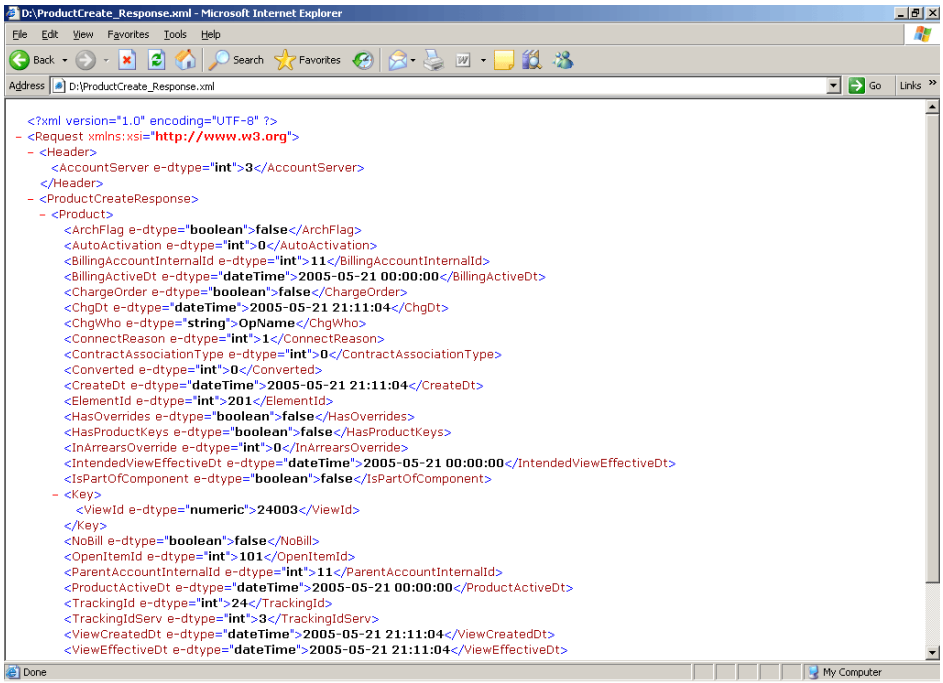
— `TrackingId`

— `TrackingIdServ`

7. Trigger the TIBCO ActiveMatrix BusinessWorksprocess. Select the **Tester** tab in the project panel and press **F9**. In the next dialog box, click **Load & Start Current**.

Expected Results

- Check the output XML file (the path of which was provided in step 5). A sample output is shown next.



- The message `Successfully processed the request` is displayed on the adapter console.

Chapter 4

TIBCO ActiveMatrix BusinessWorks: Working with the OrderedAccountCreate API Call

This chapter demonstrates the execution of the OrderedAccountCreate API Call using TIBCO ActiveMatrix BusinessWorks.

Topics

Example Description

This example demonstrates the usage of the OrderedAccountCreate API Call. The incoming request to the adapter consists of an XML message which has the `OrderedAccountCreate` node. This XML block has details based on which Kenan/BP creates a OrderedAccount. The response from Kenan/BP is also in the form of an XML message which contains the `OrderedAccountCreate` Response node.

Set up the OrderedAccountCreate Example

Task C Importing the project

Before starting the example, you must import the sample project zip file and save it in a new project.

1. Start TIBCO Designer.
   Click Programs > TIBCO > TIBCO Designer <version_num> > Designer <version_num>.
2. In the initial dialog box, click New Empty Project and specify a name for the project in the **Save Projec**t dialog. Click **OK**.
3. Import the project by clicking **Project > Import Full Project**.
4. Select the **ZIP Archive** tab in the Import dialog box.
5. Browse to select the `OrderedAccount_Create_RV.zip` file. By default, this is located in `$TIBCO_HOME/adapter/adkenan/<version_num>/examples/BusinessWorks` folder. Click **OK**.
6. In the I**mport Options** dialog box, select **Replace existing global variables with those in import** and **Overwrite on Name Conflict**. Click **Apply**.
7. For Kenan/BP 11.5 or 11.7, import the following XSD files in the order specified: `Request.xsd,Order.xsd, OrderLookup.xsd, ServiceOrder.xsd,Item.xsd, LogicalServiceOrder.xsd, AccountLocate.xsd, CreditCard.xsd, ExternalIdAcctMap.xsd, Account.xsd,AccountId.xsd, OrderAccount.xsd, OrderedAccount.xsd`. For Kenan/BP 12.0, import the following XSD files in the order specified: `Request.xsd, Order.xsd, OrderLookup.xsd, ServiceOrder.xsd, Item.xsd, LogicalServiceOrder.xsd, AccountLocate.xsd, PaymentProfile, ExternalIdAcctMap.xsd, Account.xsd,AccountId.xsd, OrderedAccount.xsd,Dependency.xsd, EntityType.xsd, Exclusion.xsd , OrderQuote.xsd`. The order is useful because `OrderedAccount.xsd` includes the schema location of the other XSD files. These XSD files are a part of the Kenan/FX installation.
8. Modify the global variables pertaining to the Security Server Settings as well as the `RvDaemon` parameter as required.
9. Save the project by selecting **Project > Save.**

Task D Converting the repository into .dat format

10. In TIBCO Designer, click **Project > Export Full Project**.

11. In the following dialog box, select the **Local Repository** tab.
12. Specify a name in the Project Name field and a valid path in the Dir Name. Click **Yes**.
13. In the following dialog box titled Create Project, click **OK**.

Running the Example

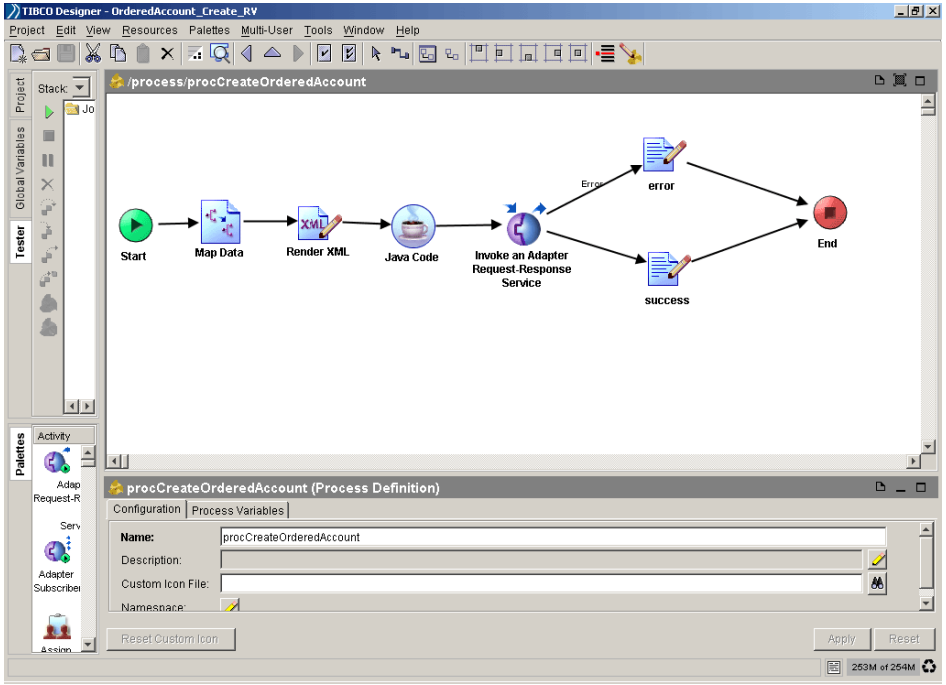
Perform the following tasks to run the example.

Task A Starting the Adapter

1. Edit the `adkenan.tra` file. This is located in `$TIBCO_HOME/adapter/adkenan/<version_num>/bin`. Specify the `repourl` as the location where the local repository was saved in the previous section (Refer Task J, Converting the repository into .dat format). Specify the `configurl` as `/tibco/private/adapter/KenanAdapterConfiguration`.
2. Open a new console session on the UNIX machine where the adapter is installed and the Kenan/FX middleware is running.
3. Move to `$TIBCO_HOME/adapter/adkenan/<version_num>/bin` and type `adkenan`. This brings up the adapter and displays the message `Successfully initialized the adapter`.

Task B Running the TIBCO ActiveMatrix BusinessWorksProcess

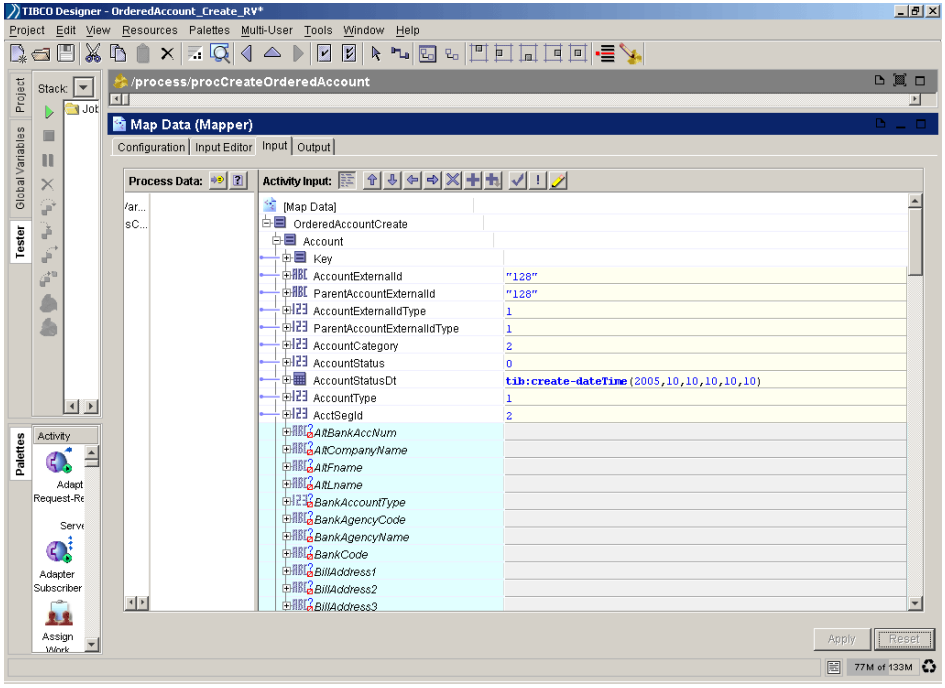
1. Start TIBCO Designer.
   Click **Programs > TIBCO > TIBCO Designer <version_num>> Designer <version_num>**.
2. In the initial dialog box, click **Open Existing Project**. Browse and select the location where the project was saved in Task A, Importing the project in the section , Set up the OrderedAccountCreate Example.
3. Click `procCreateOrderedAccount` in the `Process` folder. The BusinessWorks process is displayed as shown next:



4. Check the value or the `RvDaemon` global variable to see whether it reflects the same parameter of the repository which is being used by the adapter to run.

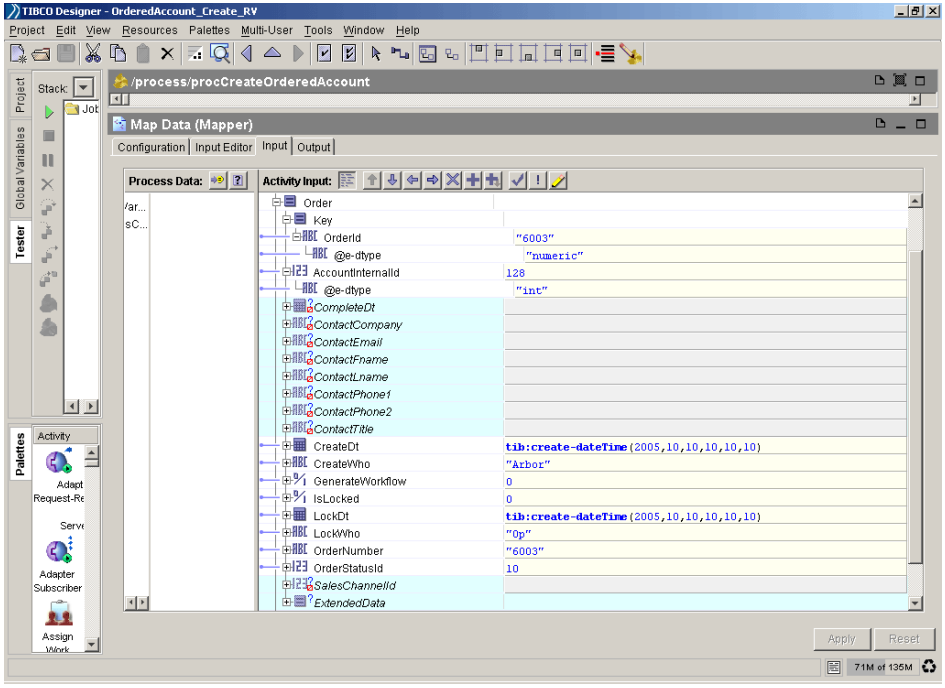
5. Provide a valid location for the output file in both the Write File activities titled **Success** and **Error**. The file associated with success will have the reply XML message sent by Kenan/BP to the request sent by the adapter and the error will have the Exception message sent by the adapter, in case of an error. To provide a valid location for the output file, double -click the corresponding **Write File** icon, select the **Input** tab and edit the filename parameter.

6. Provide a valid value for the `AccountInternalId` field in the Mapper task. Double-click the **mapper** icon, select the **Input** tab, expand the `OrderedAccountCreate` node and then the `Account` node till you see the `AccountInternalId` field (inside the Key node). Ensure that the `AccountInternalId` specified here is not present in the application records (Check the CMF table for the same). The adapter will try to create an account with the same Internal ID. You may also specify various other parameters (like `AccountExternalId, First name, Company name`) of the account to be created. You may do so by editing the parameters under the `Account` node. The configuration is displayed as shown next:



Edit the date fields `CreateDt`, `LockDt` in the order, as applicable. The format for the date fields is `tib:create-dateTime(<year>,<month>,<day>,<hr>,min>,<sec>)`. For example, 10th January, 2004 12:33:24 would be written as tib:create-dateTime(2004,01,10,12,33,24). This format is then converted by the custom Java code function to a date format acceptable to Kenan middleware. For more information about the data converter, refer to *TIBCO ActiveMatrix Adapter for Kenan/BP Configuration and Deployment*.

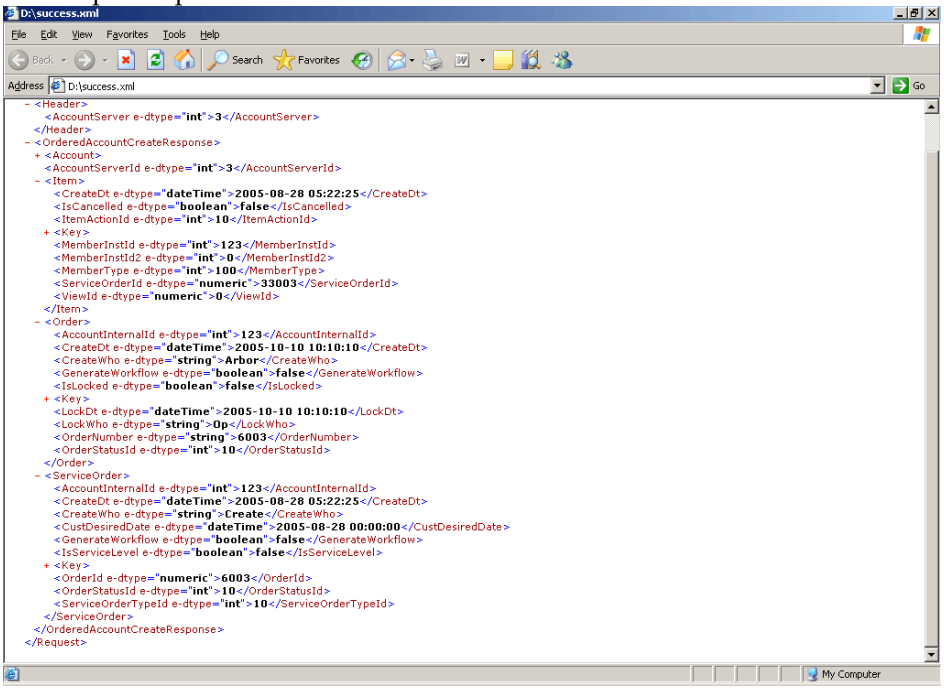
7. Provide a valid value for the Order. Double-click the **mapper** icon, select the **Input** tab, expand the `OrderedAccountCreate` node, and then `order` node till you see the OrderId field (inside the Key node). Ensure that the order is already present in the `ORD_ORDER` table and is not locked. The Mapper task is displayed as shown next:



8. Set the `VerboseResponse` flag (under the OrderedAccountCreate node) as true. This will show all the business entities created in the process.
9. Trigger the TIBCO ActiveMatrix BusinessWorksprocess. Select the **Tester** tab in the project panel and press **F9**. In the next dialog box, click **Load & Start Current**.

Expected Results

- Check the output XML file (the path of which was provided in step 5). A sample output is shown next.



- The message `Successfully processed the request` is displayed on the adapter console.

Chapter 5

TIBCO ActiveMatrix BusinessWorks: Demonstrating Custom Function Callout

This chapter demonstrates the invoking of custom functions using TIBCO ActiveMatrix BusinessWorks.

Topics

Example Description

This example demonstrates the invoking of custom functions through the adapter. Using this functionality, you can write custom code by implementing an interface provided by the adapter. Once the source code for the function has been written and compiled, the class has to be placed in the adapter class-path. Based on the incoming request, the adapter invokes the custom function and returns the response object (from the custom function) to the calling function.

In this example, the custom function creates a Connection object and uses the same to execute a Hashmap call.

Set up the Custom Function Callout Example

Task C Preparing the Custom Class

Before starting the example, you must import the sample project zip file and save it in a new project.

1. The first task is to write the custom function. This can be done by implementing the KenanMessageHandler interface exposed by the adapter. In this example, two custom classes, `AcctCustomMessageHandler.java` and `KenanCustomMessageHandler.java` have been provided in `$TIBCO_HOME/adapter/adkenan/<version_num>/examples/KenanCustomCallOut/CustomClasses`. The custom logic is to be written under the method declared as `processMessage`.
2. The next step is to prepare the custom function handler factory class. This can be done by implementing the KenanMessageHandlerFactory interface exposed by the adapter. In this example, `KenanCustomFactory.java` has been provided. The purpose of the factory class is to instantiate the custom function handler class. In this example, `KenanCustomFactory.java` has been provided in `$TIBCO_HOME/adapter/adkenan/<version_num>/examples/KenanCustomCallOut/CustomClasses`. The logic for instantiating the custom function handler has to be written under the method declared as `getMessageHandler`.

Perform the steps 1 and 2 only if you wish to create your own Java classes. The above steps are not required if you are using the Java source files provided with the adapter installation.

In case you want to use the source files provided with the adapter installation, provide a valid account internal ID in the following entry present in the `processMessage` method body:

```
acctKey.put("AccountInternalId", new Integer(1));
```

3. Once the source files of both the classes have been generated, the same have to be compiled. `setenv_customcall.sh` (in `$TIBCO_HOME/adapter/adkenan/<version_num>/examples/KenanCustomCallOut/CustomClasses`) can be used as a template for compiling the source code. Please modify the environment parameters per your settings and execute the script. This will generate the class files in the same folder.
4. The compiled class has to be placed in the classpath so that the adapter can pick the same while running. The location of the custom classes is already a part of the `CUSTOM_CP_EXT` path in the adapter properties file.

Task D Preparing the Custom XSD

The next task is to create the custom XSD (XML Schema Definition), which would be used to generate an XML message for the adapter to process. In this example, there is a schema called `CustomRequest.xsd` present in `$TIBCO_HOME/adapter/adkenan/<version_num>/examples/KenanCustomCallOut`. Two mandatory features in the XSD are:

- To have the `<xs:element name = "CustomRequest">` tag enclosing all other XML tags.
- To have an element called `CustomFunctionName`, which will contain the name of the custom function based on which the custom function factory generates the appropriate message handlers.

Task E Importing the project

Before starting the example, you must import the sample project zip file and save it in a new project.

5. Start TIBCO Designer.
   Click Programs > TIBCO > TIBCO Designer <version_num> > Designer <version_num>.
6. In the initial dialog box, click **New Empty Project** and specify a name for the project in the **Save Projec**t dialog. Click **OK**.
7. Import the project by clicking **Project > Import Full Project**.
8. Select the **ZIP Archive** tab in the Import dialog box.
9. Browse to select the `KenanCustomCallOut_Export.zip` file. By default, this is located in `$TIBCO_HOME/adapter/adkenan/<version_num>/examples/KenanCustomCallOut/Repository/BusinessWorks` folder. Click **OK**.
10. In the I**mport Options** dialog box, select **Replace existing global variables with those in import** and **Overwrite on Name Conflict**. Click **Apply**.
11. Modify the global variables pertaining to the Security Server Settings as well as the `RvDaemon` parameter as required.
12. Save the project by selecting **Project > Save.**

Task F Converting the repository into .dat format

13. In TIBCO Designer, click **Project > Export Full Project**.
14. In the following dialog box, select the **Local Repository** tab.

15. Specify a name in the Project Name field and a valid path in the Dir Name. Click **Yes**.
16. In the following dialog box titled Create Project, click **OK**.

Running the Example

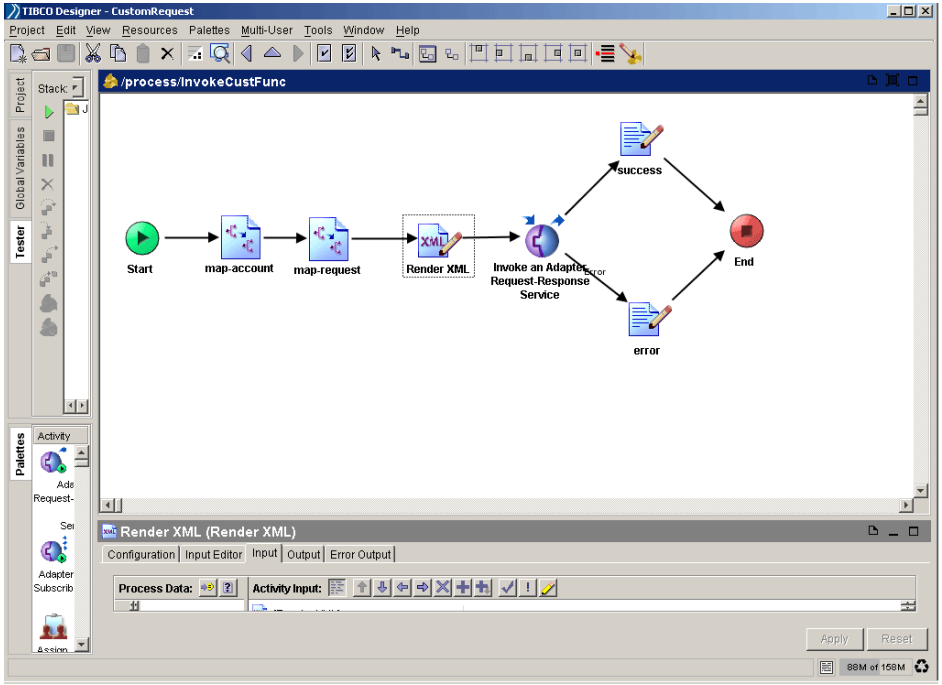
Perform the following tasks to run the example.

Task A Starting the Adapter

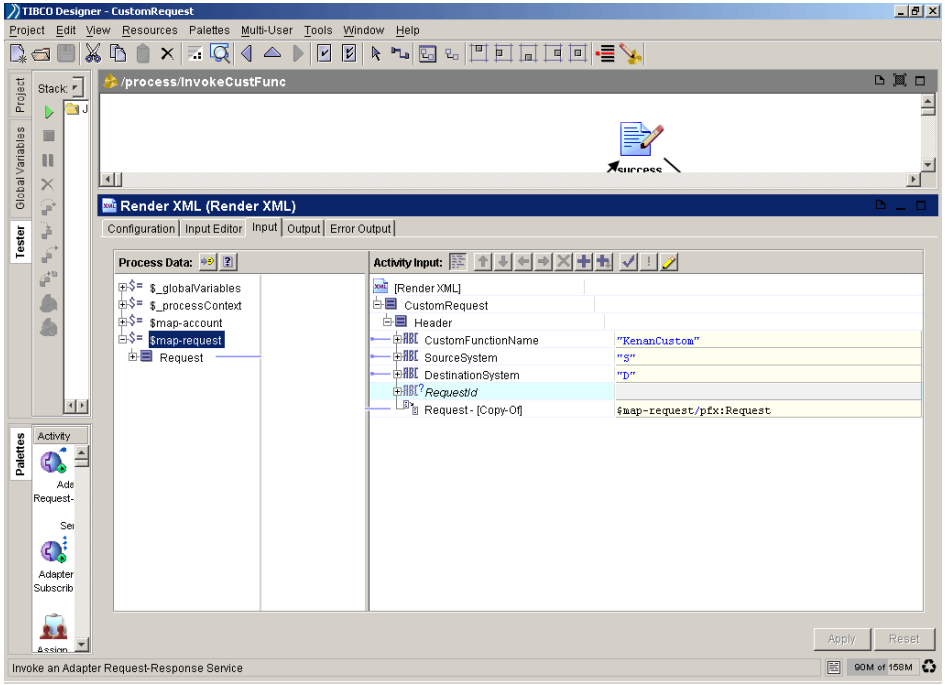
1. Edit the `adkenan.tra` file. This is located in `$TIBCO_HOME/adapter/adkenan/<version_num>/bin`. Specify the `repourl` as the location where the local repository was saved in the previous section (Refer Task J, Converting the repository into .dat format). Specify the `configurl` as `/tibco/private/adapter/KenanAdapterConfiguration`.
2. The name of the factory class needs to be specified in the adapter property file. The property `adkenan.KenanCustomCallOutFactoryName` should be uncommented and the name of Custom Factory class should be provided. For example, `adkenan.KenanCustomCallOutFactoryName KenanCustomFactory`.
3. Open a new console session on the UNIX machine where the adapter is installed and the Kenan/FX middleware is running.
4. Move to `$TIBCO_HOME/adapter/adkenan/<version_num>/bin` and type `adkenan`. This brings up the adapter and displays the message `Successfully initialized the adapter`.

Task B Running the TIBCO ActiveMatrix BusinessWorksProcess

1. Start TIBCO Designer.
   Click **Programs > TIBCO > TIBCO Designer <version_num>> Designer <version_num>**.
2. In the initial dialog box, click **Open Existing Project**. Browse and select the location where the project was saved.
3. Click `procInvokeCustFunc` in the `Process` folder. The BusinessWorks process is displayed as shown next:



4. Check the value or the `RvDaemon` global variable to see whether it reflects the same parameter of the repository which is being used by the adapter to run.
5. Provide a valid location for the output file in both the Write File activities titled **Success** and **Error**. The file associated with success will have the reply XML message sent by Kenan/BP to the request sent by the adapter and the other one (associated with error) will have the Exception message being sent by the adapter, in case of error. For providing a valid location for the output file, double-click the corresponding **Write File** icon, select the **Input** tab and edit the filename parameter.
6. Provide a valid value for the custom function. Double-click the Render XML task. Expand the `CustomRequest` node followed by the `Header` node till you reach the `CustomFunctionName` node. For the factory class in this example, the valid values of `CustomFunctionName` are `KenanCustom` and `KenanCustomAcct`. The render XML task is displayed as shown next:



7. Trigger the TIBCO ActiveMatrix BusinessWorksprocess. Select the **Tester** tab in the project panel and press **F9**. In the next dialog box, click **Load & Start Current**.

Expected Results

- A sample output is shown next.

```
Inside KenanCustomMessageHandler
Call to be made:
callName: B_AcLoGt
  AccountLocate = {
    Key = {
      AccountInternalId = '1'
    }
  }
The connection received after getHMConnection
:com.csgsystems.aruba.connection.Connection@3a633d51
The context to be used for execution is
:com.csgsystems.aruba.connection.BSDMSessionContext@494eaec9
Server response to call:
  AccountLocate = {
    Key = {
      #NAME = 'Key'
      AccountInternalId = '1'
    }
    AccountCategory = '2'
    #NAME = 'AccountLocate'
    BillFnameFind = 'REBECCA'
    BillLnameFind = 'BAYWATER'
    BillCompanyFind = 'ABCD'
    BillCompany = 'ABCD'
    BillAddress1 = '123 NEWS St.'
    ServerId = '3'
    BillLname = 'Baywater'
    AcctSegId = '0'
    BillFname = 'Rebecca'
  }
  #NAME = 'AccountLocateGetResponse'
ServerId: 3
```

- The message `Successfully processed the request` is displayed on the adapter console.

Chapter 6

TIBCO ActiveMatrix BusinessWorks: Working with the User-Defined Transaction (UDT) API Call

This chapter demonstrates the execution of the User-Defined Transaction API Call using TIBCO ActiveMatrix BusinessWorks.

Topics

Example Description

This example demonstrates the usage of the User-Defined Transactions API Call. There are two BusinessWorks processes in the example. In the process `udt`, the direct parsing feature is disabled. However, it is enabled in the process `udt (direct parse)`.

In both processes, the incoming request to the adapter consists of an XML message which has the `CustomerUdtRequest` node including the `AccountGet` and `ProductGet` nodes. The response from Kenan/BP is also an XML message which contains the `UDT Response` node with `Account` and `Product` nodes.

Set up the UDT_RV Example

Task C Importing the project

Before starting the example, you must import the sample project zip file and save it in a new project.

1. Start TIBCO Designer.
   Click Programs > TIBCO > TIBCO Designer <version_num> > Designer <version_num>.
2. In the initial dialog box, click **New Empty Project** and specify a name for the project in the Save Project dialog. Click **OK**.
3. Import the project by clicking **Project > Import Full Project**.
4. Select the **ZIP Archive** tab in the Import dialog box.
5. Browse to select the `UDT_RV.zip` file. By default, this is located in the *<TIBCO_HOME>*`\adapter\adkenan\<version_num>\examples\BusinessWorks` folder. Click **OK**.
6. In the Import Options dialog box, select **Replace existing global variables with those in import** and **Overwrite on Name Conflict**. Click **Apply**.
7. Modify the global variables pertaining to the Security Server Settings as well as the `RvDaemon` parameter as required.
8. Save the project by selecting **Project > Save**.

Task D Converting the repository into the .dat format

9. In TIBCO Designer, click **Project > Export Full Project**.
10. In the Export Project dialog box, select the **Local Repository** tab.
11. Specify a name in the Project Name field and a valid path in Dir Name. Click **Yes**.
12. In the following dialog box titled Create Project, click **OK**.

Running the Example

Perform the following tasks to run the example:

Task A Running the Adapter

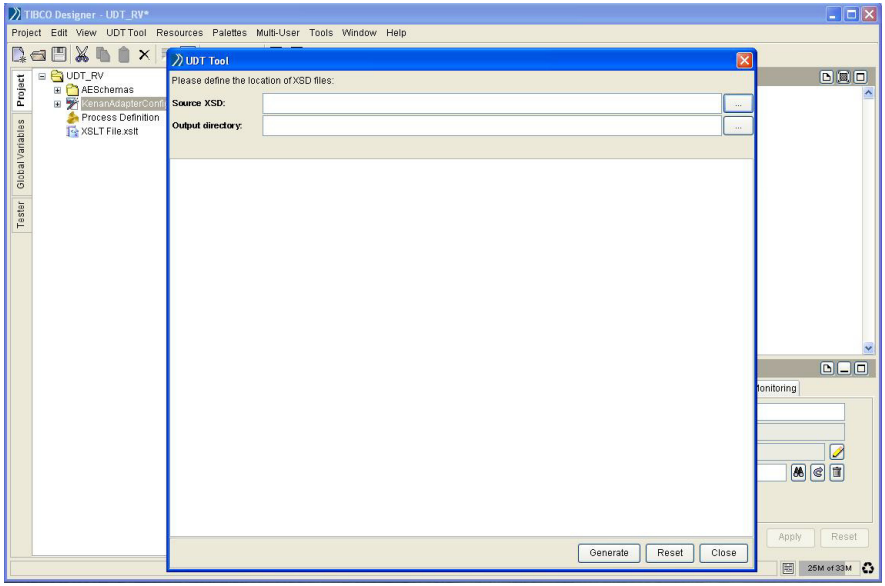
1. Put the local repository `.dat` file metioned in the previous section into any location on the UNIX machine where the adapter is installed and the Kenan/FX middleware is running.
2. Edit the `adkenan.tra` file. This is located in `$TIBCO_HOME/adapter/adkenan/<version_num>/bin`.
   - Specify the `repourl` as the location of the local repository `.dat` file mentioned in step 1.
   - Specify the `configurl` as `KenanAdapterConfiguration` if the **udt** BusinessWorks process runs, and `KenanAdapterConfiguration1` if the **udt (direct parse)** BusinessWorks process runs.
3. Open a new console session. Change the working directory to `$TIBCO_HOME/adapter/adkenan/<version_num>/bin` and type **adkenan**. Executing this command starts the adapter. If the adapter is started successfully, the following message is displayed:

   ```
   Successfully initialized the adapter.
   ```

Task B Running the TIBCO ActiveMatrix BusinessWorksProcess

4. Start TIBCO Designer.
   Click **Programs > TIBCO > TIBCO Designer <version_num> > Designer <version_num>**
5. In the initial dialog box, click **Open Existing Project**. Browse and select the location where the project was saved.

6. Click **KenanAdapterConfiguration**. Click **UDT Tool > Load UDT Tool**. The UDT Tool is displayed:



7. Provide the **Source XSD** and **Output directory** fields with values.

   a. Click the browse button opposite to the Source XSD field to locate the `CustomerUdtRequest.xsd` file which is provided by CSG. After this .xsd file is selected, the various elements are loaded and displayed in the UDT Tool window.

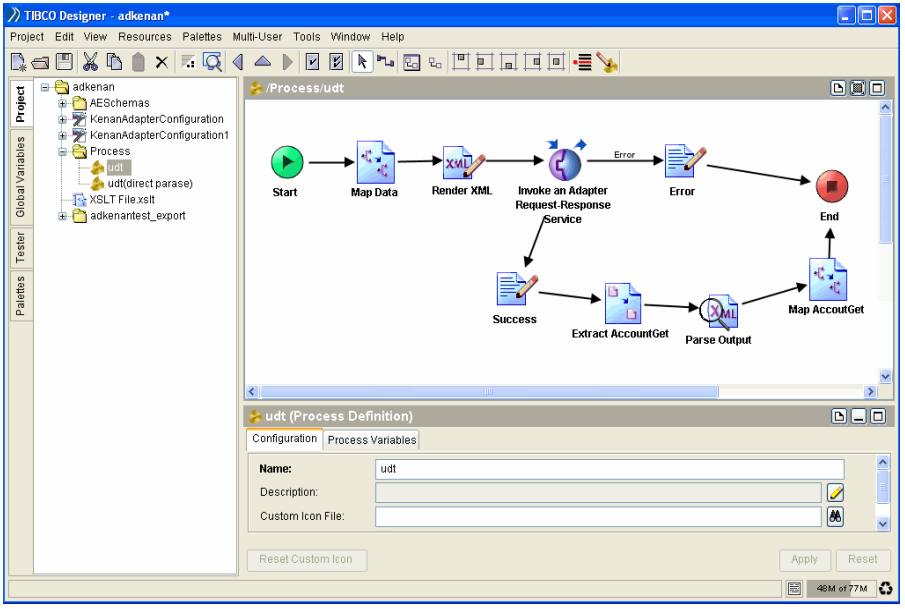
   b. Select the check boxes against the elements to include. For this example, select `Account` and `Product`.

   c. Click the browse button opposite to the Output directory field to select a folder as the Output directory.

   d. Click **Generate** in the UDT Tool window. A message box shows that 8 files are generated in the Output directory.

   e. Click **Close** to close the UDT Tool window.

8. Import the generated XSD files.
   a. Select **Project > Import Resources from File, Folder, URL**.
   b. Select **Folder** from the Format drop-down box.
   c. Click the browse button opposite to the Folder field to select the folder where the 8 .xsd fildes generated are.
   d. Click **OK**. The Import Successful message box displays. Click **OK**.
9. Edit the process definition.
   Click **udt** under the Process folder in the Project panel. The BusinessWorks process is displayed as shown next:



10. Set the `RvDaemon` global variable in the `udt` process to the value of the `RvDaemon` parameter in the repository which is used by the adapter.
11. Provide a valid location in the udt process for the output file in the Write File activitiy named Success and the one named Error.
    — The file associated with Success includes the response XML message from Kenan/BP. This response is for the request sent by the adapter to Kenan/BP.
    — The file associated with Error will include an exception message which is sent by the adapter, if any error appears.

    To provide a valid location for the output file, click the corresponding **Write File** icon, select the **Input** tab and edit the fileName parameter.

12. In the udt process, click the Mapper activity named **Map Data**. Select the Input tab and in the Activity Input, go to **CustomerUdtRequest>RequestList>Account>AccountGet>Account>Key>AccountInternalId**. Provide a valid value for the **AccountInternalId** field. The value of **AccountInternalId** specified here must be one of the values in the `ACCOUNT_NO` column of the `CMF` table.
13. In the Activity Input, go to **CustomerUdtRequest>RequestList>Product>ProductGet>Product>Key>ViewId**. The value of the ViewId specified here must be one of the values in the `VIEW_ID` column of the `PRODUCT_VIEW` table.
14. Trigger the TIBCO ActiveMatrix BusinessWorks process. Select the **Tester** tab in the project panel and press **F9**. In the next dialog box, click **Load & Start Current**. Make sure the configurl is specified as `KenanAdapterConfiguration` as described in Task A.
15. Follow the same procedure as described from step 6 to step 11 to edit, check and run the process **udt (direct parse)**. Make sure the configurl is specified as `KenanAdapterConfiguration1` as described in Task A.

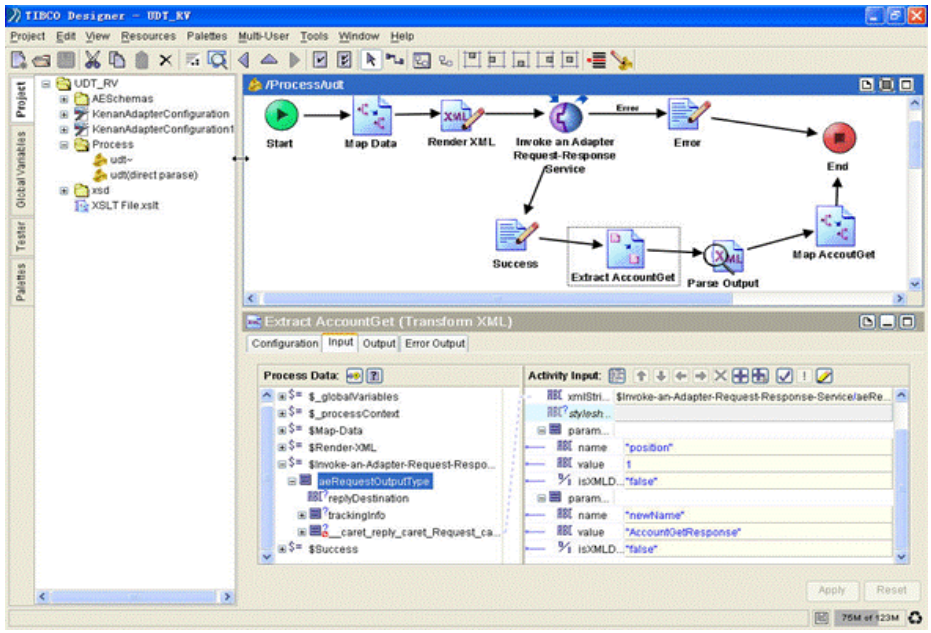
Expected Results

Expected Results for the Process udt

- When the response status returned by the activity "Invoke an Adapter Request-Response Service" in this process is Success, verify the output XML file associated with the Success activity. The location of this file is specified in Step 8 of Task B. The message includes responses of the `AccountGet` and `ProductGet` requests. These responses are wrapped by the `UDTResponse` in the `UDTResponseList`. A sample output is shown as below:

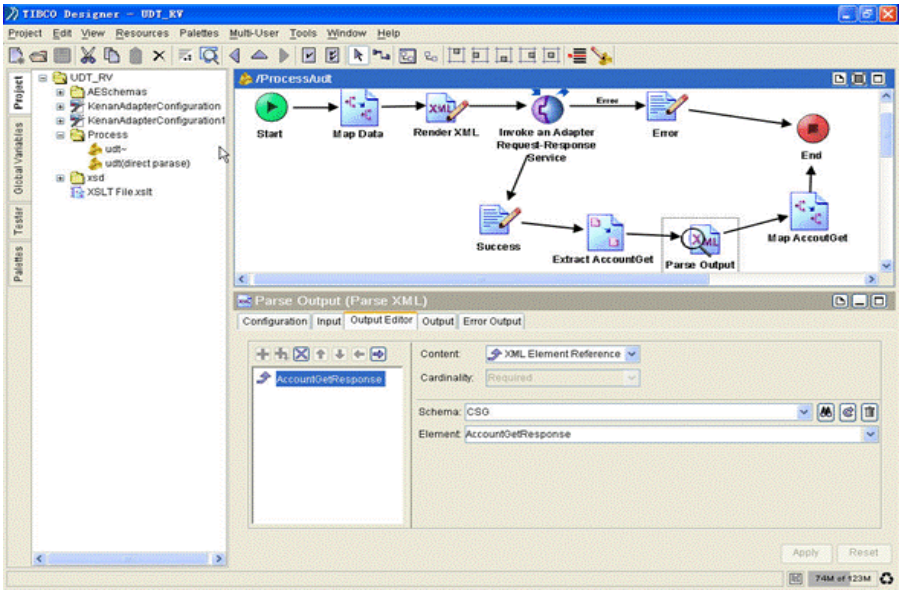
```
<?xml version="1.0" encoding="UTF-8" ?>
- <Request xmlns="CSG" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  - <Header>
      <AccountServer e-dtype="int">3</AccountServer>
      <CallCorrelation e-dtype="string">oracle.xml.parser.v2.XMLElement@4c6504bc</CallCorrelation>
    </Header>
  - <UDTResponse>
    - <UDTResponseList e-dtype="list">
      - <UDTResponse>
        + <Account>
          <RequestId e-dtype="string">AccoutGet</RequestId>
          <RequestObjName e-dtype="string">Account</RequestObjName>
        </UDTResponse>
      - <UDTResponse>
        + <Product>
          <RequestId e-dtype="string">ProductGet</RequestId>
          <RequestObjName e-dtype="string">Product</RequestObjName>
        </UDTResponse>
      </UDTResponseList>
    </UDTResponse>
</Request>
```

- The Extract AccountGet activity extracts the `AccountGetResponse` based on the XSLT File.xslt file. These configurations are shown in the next figure.



- The output of the `Extract AccountGet` activity is set to the `AccountGetResponse` element from the `CustomerUdtRequest.xsd`, which is

shown as below.



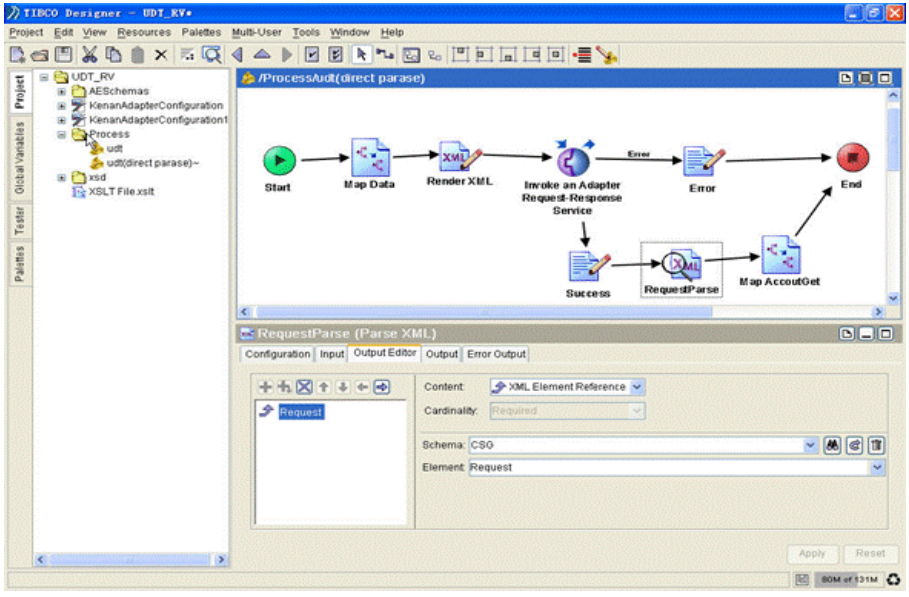
Expected Results for the Process udt (direct parse)

- Verify the output XML file for the Task Success. The message includes responses of the `AccountGet` and `ProductGet` requests. These responses are wrapped by the `UDTResponse` in the `UDTResponseList`. The

`UDTResponseList` has two nodes: `AccountGetResponse` and `ProductGetResponse`. A sample output is shown as below:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <Request xmlns="CSG" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  - <Header>
      <AccountServer e-dtype="int">3</AccountServer>
      <CallCorrelation e-dtype="string">oracle.xml.parser.v2.XMLElement@315b0333</CallCorrelation>
    </Header>
  - <UDTResponse>
    - <UDTResponseList e-dtype="list">
      - <AccountGetResponse>
        + <Account>
          <RequestId e-dtype="string">AccoutGet</RequestId>
          <RequestObjName e-dtype="string">Account</RequestObjName>
        </AccountGetResponse>
      - <ProductGetResponse>
        + <Product>
          <RequestId e-dtype="string">ProductGet</RequestId>
          <RequestObjName e-dtype="string">Product</RequestObjName>
        </ProductGetResponse>
      </UDTResponseList>
    </UDTResponse>
  </Request>
```

- The RequestParse activity can directly parses the response data from the adapter. The configuration for this activity is shown as below:



Chapter 7

TIBCO ActiveMatrix BusinessWorks: Working with the Orders Services API Call

This chapter demonstrates the execution of the Orders Services API Call using TIBCO ActiveMatrix BusinessWorks.

Topics

Example Description

This example demonstrates the usage of the Orders Services API Call. The sample includes following steps:

1. **createAccount**

   Create an Account and insert it into the database.

2. **createOrder**

   Create an Order and associate it with the Account.

3. **createServiceOrder**

   Create an Account ServiceOrder, associate it with the Account and Order, and insert it into the database.

4. **createAccountLevelProduct**

   Create an Account-level Product, associate it with the Account, and insert it into the database.

5. **createItemforProduct**

   Create an Item for the Account-level Product, associate it with the Account ServiceOrder and Account-level Product, and insert it into the database.

6. **commitOrder**

   Commit the Order. If this step is changed to cancelOrder, the Account Level Product created in step 4 will not get created for this account.

Set up the Orders Services Example

Task C Importing the project

Before starting the example, you must import the sample project zip file and save it in a new project.

7. Start TIBCO Designer.
   Click **Programs > TIBCO > TIBCO Designer <version_num>> Designer <version_num>**.
8. In the initial dialog box, click New Empty Project and specify a name for the project in the Save Project dialog. Click **OK**.
9. Import the project by clicking **Project > Import Full Project**.
10. Select the **ZIP Archive** tab in the Import dialog box.
11. Browse to select the `Order_Service_RV.zip` file. By default, this is located in the `$TIBCO_HOME/adapter/adkenan/<version_num>/examples/BusinessWorks` folder. Click **OK**.
12. In the Import Options dialog box, select Replace existing global variables with those in import and Overwrite on Name Conflict. Click **Apply**.
13. For Kenan/BP 11.5 or 11.7, import the following XSD files: `Account.xsd, AccountId.xsd, AccountLocate.xsd, Address.xsd, AddressHistory.xsd, CreditCard.xsd, CustomerIdEquipMap.xsd, ExternalIdAcctMap.xsd, Item.xsd, LogicalServiceOrder.xsd, Order.xsd, OrderedAccount.xsd, OrderedService.xsd, OrderLookup.xsd, Product.xsd, request.xsd, Service.xsd, ServiceOrder.xsd`. For Kenan/BP 12.0, import the following XSD files: `Account.xsd, AccountId.xsd, AccountLocate.xsd, Address.xsd, PaymentProfile.xsd, CustomerIdEquipMap.xsd, ExternalIdAcctMap.xsd, Item.xsd, LogicalServiceOrder.xsd, Order.xsd, OrderedAccount.xsd, OrderedService.xsd, OrderLookup.xsd, Product.xsd, request.xsd, Service.xsd, ServiceOrder.xsd, ServiceAddressAssoc.xsd` . These XSD files are a part of the Kenan/FX installation. To import an XSD, select **Project > Import Resources from File, Folder, URL**. Select the file (`.xsd`, `.xslt`, `.wsdl`) from the drop-down box and browse to point to the XSD location.
14. Modify the global variables pertaining to the Security Server Settings as well as the `RvDaemon` parameter as required.
15. Save the project by selecting **Project > Save**.

Task D Converting the repository into .dat format

16. In TIBCO Designer, click **Project > Export Full Project**.
17. In the following dialog box, select the **Local Repository** tab.
18. Specify a name in the Project Name field and a valid path in the Dir Name. Click **Yes**.
19. In the following dialog box, titled Create Project, click **OK**.

Running the Example

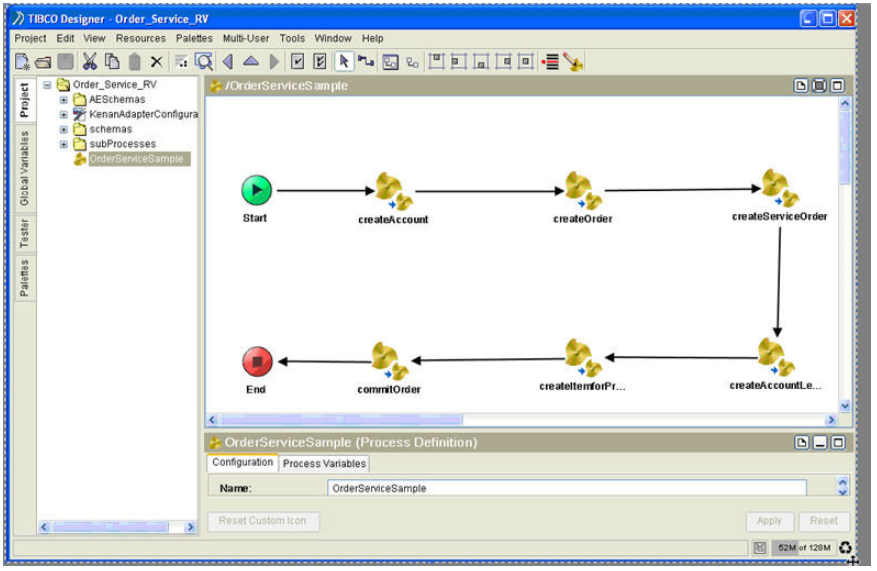
Perform the following tasks to run the example.

Task E Starting the Adapter

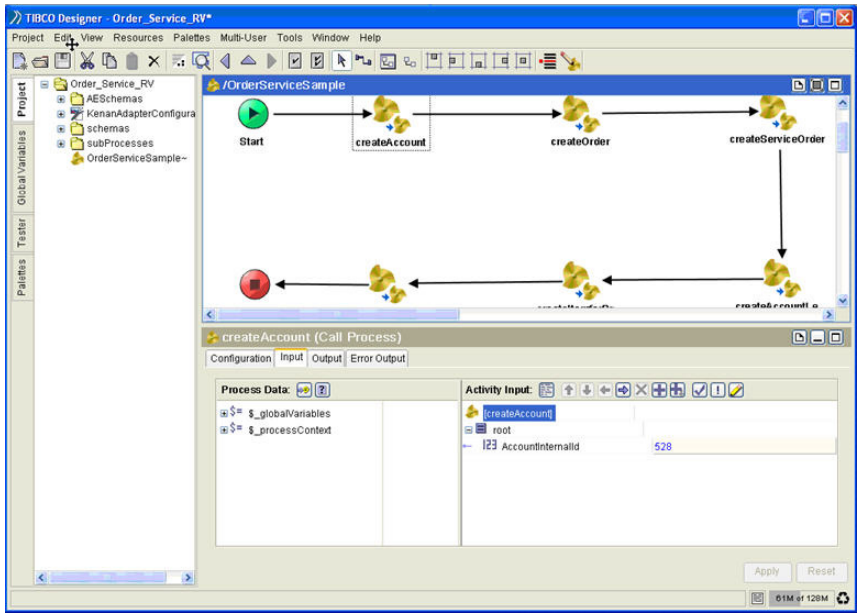
1. Edit the `adkenan.tra` file. This file is located in `$TIBCO_HOME/adapter/adkenan/<version_num>/bin`.
   - — Specify the `repourl` as the location where the local repository was saved in the previous section.
   - — Specify the `configurl` as `/tibco/private/adapter/KenanAdapterConfiguration`.
2. Open a new console session on the UNIX machine where the adapter is installed and the Kenan/FX middleware is running.
3. Move to `$TIBCO_HOME/adapter/adkenan/<version_num>/bin` and type **adkenan**. This brings up the adapter and displays the message `Successfully initialized the adapter`.

Task F Running the TIBCO ActiveMatrix BusinessWorksProcess

4. Start TIBCO Designer.
   Click Programs > TIBCO > TIBCO Designer <version_num> > Designer <version_num>.
5. In the initial dialog box, click **Open Existing Project**. Browse and select the location where the project was saved.
6. Click `OrderServiceSample`. The `OrderServiceSample` process is displayed as shown next:



7. Ensure the value of the `RvDaemon` global variable matches the value of the `RvDaemon` parameter in the repository (which is being used by the adapter to run).
8. Provide a valid value for the `AccountInternalId` field in the createAccount task. Double-click the createAccount task, select the **Input** tab, expand the root node and then you see the `AccountInternalId` field. Ensure that the `AccountInternalId` specified here is not present in the application records (Check the CMF table for the same). You can also drill down every sub process (for example, `createOrder`, `createServiceOrder`) to configure parameters as needed. The configuration is displayed as shown next:



Edit the date field CreateDt in the ItemCreate, as applicable. The format for the date field is <year>-<month>-<day>T<hr>:<min>:<sec>. For example, `2nd April 2007 10:30 am` is written as `2007-04-02T10:30:00`. This format is then converted by the custom Java code function to a date format acceptable to the Kenan middleware. For more information about the data converter, refer to *TIBCO ActiveMatrix Adapter for Kenan/BP Configuration and Deployment*.

9. Trigger the TIBCO ActiveMatrix BusinessWorks process. Select the **Tester** tab in the project panel and press **F9**. In the next dialog box, click **Load & Start Current.**

Expected Results

- **Account** and **Account-level Product** are created.
- The message `Successfully processed the request` is displayed on the adapter console.

Index