BEA WebLogic Integration Adapter for Manugistics®

User Guide

Version 8.1.1
Document Revised: October 2003

Copyright

Restricted Rights Legend

Trademarks or Service Marks

Contents

About This Document

1. Introducing the BEA WebLogic Adapter for Manugistics

2. Generating Schemas for Manugistics Integration Objects

3. Defining Application Views for Manugistics

About This Document

This document describes how to use the BEA WebLogic Adapter for Manugistics. This document is organized as follows:

- Chapter 1, "Introducing the BEA WebLogic Adapter for Manugistics," describes the adapter, how it relates to both Manugistics business objects and WebLogic Integration.
- Chapter 2, "Generating Schemas for Manugistics Integration Objects," describes how to generate schemas for your Manugistics business objects using the BEA Application Explorer.
- Chapter 3, "Defining Application Views for Manugistics," describes application views and how to use them to configure events and services.

Who Should Read This Documentation

This document is intended for the following members of an integration team:

- Integration Specialists—Lead the integration design effort. Integration specialists have expertise in defining the business and technical requirements of integration projects, and in designing integration solutions that implement specific features of WebLogic Integration. The skills of integration specialists include business and technical analysis, architecture design, project management, and WebLogic Integration product knowledge.
- Technical Analysts—Provide expertise in an organization's information technology infrastructure, including telecommunications, operating systems, applications, data repositories, future technologies, and IT organizations. The skills of technical analysts include technical analysis, application design, and information systems knowledge.

- Enterprise Information System (EIS) Specialists—Provide domain expertise in the systems that are being integrated using WebLogic Integration adapters. The skills of EIS specialists include technical analysis and application integration design.
- System Administrators—Provide in-depth technical and operational knowledge about databases and applications deployed in an organization. The skills of system administrators include capacity and load analysis, performance analysis and tuning, deployment topologies, and support planning.

Additional Information

To learn more about the software components associated with the adapter, see the following documents:

- *BEA WebLogic Adapter for Manugistics Release Notes*

  http://edocs.bea.com/wladapters/manugistics/docs811/pdf/relnotes.pdf
- *BEA WebLogic Adapter for Manugistics Installation and Configuration Guide*

  http://edocs.bea.com/wladapters/manugistics/docs811/pdf/install.pdf
- *Introduction to the BEA WebLogic Adapters*

  http://edocs.bea.com/wladapters/docs81/pdf/intro.pdf
- BEA WebLogic Adapters 8.1 Dev2Dev Product Documentation

  http://dev2dev.bea.com/products/wladapters/index.jsp
- Application Integration documentation

  http://edocs.bea.com/wli/docs81/aiover/index.html

  http://edocs.bea.com/wli/docs81/aiuser/index.html
- BEA WebLogic Integration documentation

  http://edocs.bea.com/wli/docs81/index.html
- BEA WebLogic Platform documentation

  http://edocs.bea.com/platform/docs81/index.html
- Manugistics documentation

  http://www.manugistics.com

How to Use This Document

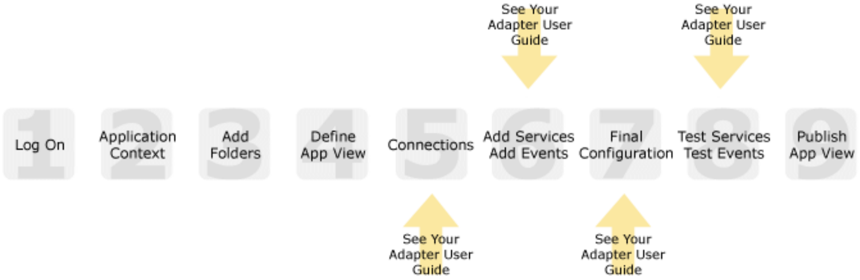
This document is designed to be used in conjunction with *Using the Application Integration Design Console*, available at the following URL:

http://edocs.bea.com/wli/docs81/aiuser/index.html

Using the Application Integration Design Console descibes, in detail, the process of defining an application view, which is a key part of making an adapter available to process designers and other users. What *Using the Application Integration Design Console* does *not* cover is the specific information about Adapter for EIS that you need to supply to complete the application view definition. You will find that information in this document.

At each point in *Using the Application Integration Design Console* where you need to refer to this document, you will see a note that directs you to a section in your adapter user guide, with a link to the edocs page for adapters. The following roadmap illustration shows where you need to refer from *Using the Application Integration Design Console* to this document.

Figure 1 Information Interlock with *Using the Application Integration Design Console*


See Your Adapter User Guide
See Your Adapter User Guide
Log On
Application Context
Add Folders
Define App View
Connections
Add Services Add Events
Final Configuration
Test Services Test Events
Publish App View
See Your Adapter User Guide
See Your Adapter User Guide

Contact Us!

Your feedback on the BEA WebLogic Adapter for Manugistics documentation is important to us. Send us e-mail at **docsupport@bea.com** if you have questions or comments. Your comments will be reviewed directly by the BEA professionals who create and update the BEA WebLogic Adapter for Manugistics documentation.

In your e-mail message, please indicate that you are using the documentation for BEA WebLogic Adapter for Manugistics and the version of the documentation.

If you have any questions about this version of BEA WebLogic Adapter for Manugistics, or if you have problems using the BEA WebLogic Adapter for Manugistics, contact BEA Customer Support through BEA WebSUPPORT at **www.bea.com**. You can also contact Customer Support by using the contact information provided on the Customer Support Card which is included in the product package.

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type
- The name and version of the product you are using
- A description of the problem and the content of pertinent error messages

Documentation Conventions

The following documentation conventions are used throughout this document.

Convention	Item
boldface text	Indicates terms defined in the glossary.
Ctrl+Tab	Indicates that you must press two or more keys simultaneously.
italics	Indicates emphasis or book titles.

`monospace text`	Indicates code samples, commands and their options, data structures and their members, data types, directories, and file names and their extensions. Monospace text also indicates text that you must enter from the keyboard.
	Examples:
	`#include <iostream.h> void main ( ) the pointer psz`
	`chmod u+w *`
	`\tux\data\ap`
	`.doc`
	`tux.doc`
	`BITMAP`
	`float`

`monospace boldface text`	Identifies significant words in code. *Example*: `void` **`commit`** `( )`
`monospace italic text`	Identifies variables in code. *Example*: `String` *`expr`*
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. *Examples*: LPT1 SIGNON OR
`{ }`	Indicates a set of choices in a syntax line. The braces themselves should never be typed.

Convention	Item

`[ ]`	Indicates optional items in a syntax line. The brackets themselves should never be typed.
	Example:
	`buildobjclient [-v] [-o name ] [-f file-list]...`
	`[-l file-list]...`

|	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.

`...`	Indicates one of the following in a command line:
	• That an argument can be repeated several times in a command line
	• That the statement omits additional optional arguments
	• That you can enter additional parameters, values, or other information
	The ellipsis itself should never be typed.
	Example:
	`buildobjclient [-v] [-o name ] [-f file-list]...`
	`[-l file-list]...`

. . .	Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.

Introducing the BEA WebLogic Adapter for Manugistics

This section introduces the BEA WebLogic Adapter for Manugistics and describes how the adapter enables integration with Manugistics business objects and WebLogic Integration.

It includes the following topics:

- About the BEA WebLogic Adapter for Manugistics
- Getting Started With the Adapter for Manugistics

About the BEA WebLogic Adapter for Manugistics

The BEA WebLogic Adapter for Manugistics connects to your Manugistics system so that you can easily use your Manugistics data and functions within your business processes. The adapter provides scalable, reliable, and secure access to your Manugistics system.

This section includes the following topics:

- Supported Manugistics Operations for Application Integration
- Supported Services
- Supported Events
- Benefits of the Adapter for Manugistics

Supported Manugistics Operations for Application Integration

The Adapter for Manugistics supports synchronous and asynchronous, bi-directional message interactions for Manugistics NetWORKS Demand Planning, Fulfillment and Transport modules.

It provides integration with the following Manugistics operations:

- Access to Manugistics supply chain management solutions and transportation management system using XML to handle both services and events
- Direct invocation of Manugistics business services, workflows, and business components
- Use of the Manuba batch process to import data into and export data from the Manugistics database or Manugistics universal data model (UDM)

Supported Services

The Adapter for Manugistics supports requests to and responses from Manugistics with services. To invoke a service, the adapter sends a request to Manugistics to cause Manugistics to perform a function. Service requests usually have responses.

Supported Events

The Adapter for Manugistics supports the receipt of notifications from Manugistics with events. To receive an event, the adapter picks up an event XML file and passes it to an event variable within a business process.

Benefits of the Adapter for Manugistics

The combination of the adapter and WebLogic Integration supplies everything you need to integrate your workflows and enterprise applications with your Manugistics system. The Adapter for Manugistics provides these benefits:

- Support for the Manugistics supply chain management solutions and the Manugistics Transportation Management system.
- Integration can be achieved without custom coding.
- Business processes can be started by events generated by Manugistics.
- Business processes can request and receive data from your Manugistics system using services.

- Adapter events and services are standards-based. The adapter services and events provide extensions to the *J2EE Connector Architecture* (JCA) version 1.0 from Sun Microsystems, Inc. For more information, see the Sun JCA page at the following URL:

  `http://java.sun.com/j2ee/connector/`

- The adapter and WebLogic Integration solution is scalable. The BEA WebLogic Platform provides clustering, load balancing, and resource pooling for a scalable solution. For more information about scalability, see the following URL:

  `http://edocs.bea.com/wls/docs81/cluster/index.html`

- The adapter and WebLogic Integration solution benefits from the fault-tolerant features of the BEA WebLogic Platform. For more information about high availability, see the following URL:

  `http://edocs.bea.com/wli/docs81/deploy/index.html`

- The adapter and WebLogic Integration solution is secure, using the security features of the BEA WebLogic Platform and the security of your Manugistics system. For more information about security, see the following URL:

  `http://edocs.bea.com/wls/docs81/secintro/index.html`

Getting Started With the Adapter for Manugistics

This section gives an overview of how to get started using the BEA WebLogic Adapter for Manugistics within the context of an application integration solution. Integration with Manugistics involves the following tasks:

- Step 1: Design the Application Integration Solution
- Step 2: Determine the Required Manugistics Business Workflows
- Step 3: Generate Schemas for Manugistics Integration Objects
- Step 4: Define Application Views and Configure Services and Events
- Step 5: Integrate with Other BEA Software Components
- Step 6: Deploy the Solution to the Production Environment

Step 1: Design the Application Integration Solution

The first step is to design an application integration solution, which includes (but is not limited to) such tasks as:

- Defining the overall scope of application integration.
- Determining the business process(es) to integrate.
- Determining which WebLogic Platform components will be involved in the integration, such as web services or workflows designed in WebLogic Workshop, portals created in WebLogic Portal, and so on.
- Determining which external systems and technologies will be involved in the integration, such as Manugistics systems and other EISs.
- Determining which BEA WebLogic Adapters for WebLogic Integration will be required, such as the BEA WebLogic Adapter for Manugistics. An application integration solution can involve multiple adapters.

This step involves the expertise of business analysts, system integrators, and EIS specialists (including Manugistics specialists). Note that an application integration solution can be part of a larger integration solution.

Step 2: Determine the Required Manugistics Business Workflows

Within the larger context of an application integration project, you must determine which specific Manugistics integration objects and workflows are required for services and events to support the business processes in the application integration solution.

Factors to consider include (but are not limited to):

- Type of Manugistics integration objects, and transport used to access the Manugistics system.
- Manugistics transactions involved in business processes.
- Logins required to access Manugistics transports and perform the required operations.
- Whether operations are, from the adapter point of view:
  - services, which notify the Manugistics system with a request for action, and, in addition, whether such services should be processed synchronously or asynchronously.
  - events, which are notifications from the Manugistics system that trigger workflows.

This step involves the expertise of Manugistics specialists, including analysts and administrators.

Step 3: Generate Schemas for Manugistics Integration Objects

After identifying the Manugistics integration objects and workflows required for the application integration solution, you must generate the XML schemas that will be used to exchange data with one or more Manugistics systems:

- Services require two XML schemas: one for the Manugistics request and another for the Manugistics response.
- Events require a single XML schema to handle the data sent by the Manugistics system.

To learn more about schemas, see Chapter 2, "Generating Schemas for Manugistics Integration Objects."

Step 4: Define Application Views and Configure Services and Events

After you create the schemas for your Manugistics services or events, you create an application view that provides an XML-based interface between WebLogic Server and a particular Manugistics system within your enterprise. If you are accessing multiple Manugistics systems, you define a separate application view for each Manugistics system you want to access. To provide different levels of security access (such as "guest" and "administrator"), define a separate application view for each security level.

Once you define an application view, you can configure events and services in that application view that employ the XML schemas that you created in "Step 3: Generate Schemas for Manugistics Integration Objects". To learn more about generating schemas, see Chapter 2, "Generating Schemas for Manugistics Integration Objects."

To learn more about defining application views, see Chapter 3, "Defining Application Views for Manugistics" in conjunction with *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

Step 5: Integrate with Other BEA Software Components

Once you have configured and published one or more application views for Manugistics integration, you can integrate these application views into other BEA software components, such as workflows or web services created in BEA WebLogic Workshop, or portals built with BEA WebLogic Portal.

For more information, see *Using the Application Integration Design Console*, particularly Chapter 3, "Using Application Views with Application Workflows," at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

Step 6: Deploy the Solution to the Production Environment

After you have designed, built, and tested your application integration solution, you can deploy it into a production environment. The following list describes some of the tasks involved in deploying an application integration:

- Design the deployment.
- Deploy the required components of the BEA WebLogic Platform.
- Install and deploy the BEA WebLogic Adapter for Manugistics as described in the *BEA WebLogic Adapter for Manugistics Installation and Configuration Guide*.
- Deploy your application views and schemas for Manugistics integration.
- Verify business processes in the production environment.
- Monitor and tune the deployment.

CHAPTER 2

Generating Schemas for Manugistics Integration Objects

The Adapter for Manugistics uses XML documents to communicate with your Manugistics system's integration objects for both services and events. Because Manugistics doesn't expose any of the metadata the BEA Application Explorer uses to generate XML schemas, you must create these schemas by running utilities provided in your adapter distribution. The utilities create service and event schemas for both SCPO and Transportation databases.

This section explains how to generate schemas. It contains the following topics:

- Before You Begin
- About Schemas and Repositories
- Defining Schemas
- Next Steps

Before You Begin

Before you begin to generate schema for the Adapter for Manugistics, you must:

- Obtain and install the BEA WebLogic Adapter for Manugistics. To learn more about this, see the *BEA WebLogic Adapter for Manugistics Installation and Configuration Guide*.
- Obtain the information necessary to connect to your Manugistics system. Contact your Manugistics administrator for this information.

About Schemas and Repositories

You describe all the documents entering and exiting your WebLogic Integration system using W3C XML schemas. These schemas describe each event arriving at and propagating from the adapter, as well as each request sent to and each response received from the adapter.

Use the Application View Console to access events and services and to assign a schema to each event, request and response. You assign each application view to a schema repository; several application views can be assigned to the same repository.

All BEA WebLogic adapters use a schema repository to store their schema information for use by the Application View Console. The schema repository is a directory containing:

- a manifest file that describes the event and service schemas
- the corresponding schema descriptions

Types of Schemas

Each service or event the Adapter for Manugistics uses must be defined by a schema. You must generate XML schemas for:

- Service Requests
- Service Responses
- Events

Service Requests

Service requests are requests for action that your application makes to your Manugistics system. Requests are defined by request schema. As part of the definition, the request schema defines the input parameters required by the Manugistics system. The Manugistics system responds to the request with a service response.

Service Responses

Service responses are the way the adapter receives Manugistics response to a service request. A service response schema defines this service response. Service requests always have corresponding responses.

Events

Events are generated by the Manugistics system as a result of activity on that system. You can use these events to trigger an action in your application. For example, the Manugistics system may generate an event when customer information is updated. If your application must do something when this happens, your application is a consumer of this event. Events are defined by event schema.

Defining Schemas

The process for defining XML schemas includes the following steps:

- Naming a Schema Repository
- Creating a Manifest
- Creating a Schema

Naming a Schema Repository

The schema repository has a three-part naming convention:

Windows: `session_base_directory\adapter_type\connection_name`

UNIX: `session_base_directory/adapter_type/connection_name`

Here:

- `session_base_directory` is the schema's base session path, which represents a folder where schemas from multiple sessions can be held
- `adapter_type` is the type of adapter (for example, Manugistics, MQSeries, or SAP)
- `connection_name` is a name representing a particular instance of the adapter type. For example, ManugisticsTest may be a test system, and ManugisticsProd may be a production system.

For example, if the session path is `/usr/opt/bea`, the adapter type is Manugistics, and the connection name is ManugisticsProd, then the schema repository is this directory:

```
/usr/opt/bea/mgistics/ManugisticsProd
```

Creating a Manifest

Each schema repository has a manifest that describes the repository and its schemas. This repository manifest is stored as an XML file named `manifest.xml`.

Note: If you manually create an XML schema, the namespace prefix in the manifest must be `xsd:`

The following is a sample manifest file:

Listing 2-1 Sample manifest file

```
<manifest>
   <connection>
      <user>stsc</user>
      <password></password>
      <url>jdbc:oracle:thin:@localhost:1521:msam</url>
      <dll>ocijdbc8</dll>
      <schema>STSC</schema>
      <driver>oracle.jdbc.driver.OracleDriver</driver>
      <protocol>thin</protocol>
   </connection>
   <schemaref name="IWY_FCST">
      <event root="MANU" file="event_IWY_FCST.xsd" sql="SELECT * FROM 
IWY_FCST"/>
   </schemaref>
   <schemaref name="DEMANDPLANNING_HISTORY">
      <request root="DEMANDPLANNING" 
file="service_DEMANDPLANNING_HISTORY.xsd"/>
      <response root="DEMANDPLANNING" 
file="service_DEMANDPLANNING_HISTORY_response.xsd"/>
   </schemaref>
   <schemaref name="DEMANDPLANNING_FORECAST">
      <event root="DEMANDPLANNING" 
file="event_DEMANDPLANNING_FORECAST.xsd"/>
   </schemaref>

</manifest>
```

The manifest has a connection section and a schema reference section, named `schemaref`. The connection section contains default values that are used when creating the application view. The schema reference name appears in the schema drop-down list on the Add Service and Add Event screens in the Application View Console. Each named schema reference can contain three schemas, one for each of the schema types described in Types of Schemas.

The repository manifest is an XML file with the root element and two sub-elements:

- `connection` appears once and is optional for the BEA WebLogic Adapter for Manugistics. If you do not want to enter default values, then include only the XML tags with no values.

  Note: Do not include the password in the manifest file. Instead, set the password fields when you set the properties of the each service and event.

- `schemaref` appears multiple times, once for each schema name.

To create a manifest:

1. Create an XML file with the following structure:

```
<manifest>
   <connection>
      <user>stsc</user>
      <password></password>
      <url>jdbc:oracle:thin:@localhost:1521:msam</url>
      <dll>ocijdbc8</dll>
      <schema>STSC</schema>
      <driver>oracle.jdbc.driver.OracleDriver</driver>
      <protocol>thin</protocol>
   </connection>
.
.
.
</manifest>
```

2. For each new event or service schema you define, create a schemaref section using the following model:

```
<schemaref name="IWY_FCST">
      <event root="MANU" file="event_IWY_FCST.xsd"/>
   </schemaref>
   <schemaref name="DEMANDPLANNING_HISTORY">
      <request root="DEMANDPLANNING"
file="service_DEMANDPLANNING_HISTORY.xsd"/>
      <response root="DEMANDPLANNING"
file="service_DEMANDPLANNING_HISTORY_response.xsd"/>
   </schemaref>
   <schemaref name="DEMANDPLANNING_FORECAST">
```

```
      <event root="DEMANDPLANNING"
file="event_DEMANDPLANNING_FORECAST.xsd"/>
   </schemaref>
```

Here, the value you assign to:

- `file` is the name of the file in the schema repository
- `root` is the name of the root element in the actual instance documents that arrive at, or are sent to, the adapter. Set the root element as follows:
  - `MANU` for SQL events
  - *`desktop_name`* for SCPO batch events
  - *`interface_name`* for transport events

Creating a Schema

Schemas describe the rules of the XML documents that traverse WebLogic Integration. You can generate schemas for SCPO and Transportation services and events by using schema-generation utilities provided in your adapter distribution. Two utilities are provided: one for creating schemas matching Transportation module data and one for creating schemas matching SCPO data.

Creating schemas includes the following tasks:

1. Extracting the Schema-Generation Utilities
2. Editing the Schema-Generation Utility Script Files
3. Running the Schema-Generation Utilities

Extracting the Schema-Generation Utilities

The schema-generation utilities and other files must be extracted from the `BEA_MGISTICS_8_1.ear` file before you can use the utilities.

To extract the utilities:

1. Create a parent directory in which to place the files.

   Create the following directory:

   For Windows: *`bea_home`*`\adapters\mgistics`

   For UNIX: *`bea_home`*`/adapters/mgistics`

Here, *bea_home* represents the BEA Home directory specified for your WebLogic installation.

2. Extract the `BEA_MGISTICS_8_1.manifest.zip` file (for Windows systems) or the `BEA_MGISTICS_8_1.manifest.tar` file (for UNIX systems) from the `BEA_MGISTICS_8_1.ear` file to the directory created in step 1.

   On UNIX systems, the `tar -xvf BEA_MGISTICS_8_1.ear` command extracts all the files from the EAR file.

3. Extract the following files from the `BEA_MGISTICS_8_1.manifest.zip` or the `BEA_MGISTICS_8_1.manifest.tar` file as appropriate for your system:

   For Windows:

   - `manu_scpo_schema.bat`
   - `manu_trns_schema.bat`

   For UNIX:

   - `manu_scpo_schema.sh`
   - `manu_trns_schema.sh`

   On Windows systems, a `\manu_schema` subdirectory is created automatically within the directory to which you extract the files.

   On UNIX systems, issue the following command:

   ```
   tar -xvf BEA_MGISTICS_8_1.manifest.tar
   ```

   This creates a `/manu_schema` subdirectory within the directory to which the files were extracted. For example, if you extract the files into `bea_home/adapters/mgistics`, the files are placed in `bea_home/adapters/mgistics/manu_schema`.

   Note: A second directory, `/manu_trns_load`, is also created. For more information on the files placed in this directory, see the *BEA WebLogic Adapter for Manugistics Installation and Configuration Guide*.

4. Extract the `ibi-edaqm.jar` file from the `BEA_MGISTICS_8_1.ear` into the `manu_schema` directory.

Editing the Schema-Generation Utility Script Files

You must edit the schema-generation utility script files to run on your system.

To edit the utility script files:

1. Open the appropriate file using an ASCII editor:

For Windows systems:

- If you are creating schemas for a SCPO database open the `manu_scpo_schema.bat` file.
- If you are creating schemas for a Transportation database, open the `manu_trns_schema.bat` file.

For UNIX systems:

- If you are creating schemas for a SCPO database open the `manu_scpo_schema.sh` file.
- If you are creating schemas for a Transportation database, open the `manu_trns_schema.sh` file.

2. Edit the following statements by replacing the placeholder values as follows:

Table 2-1 Table of Parameters for Manugistics Schema Files

Statement	Replace	With
`set LIB_PATH=<my-lib-path>`	`<my-lib-path>`	The directory in which you placed the batch file. For example, `bea_home\adapters\mgistics\manu_schema`
`set JAR_FILES=%JAR_FILES%;<oracle-odbc-driver>`	`<oracle-odbc-driver>`	The path to the Oracle JDBC driver. For example, on Windows, `C:\bea\weblogic81\server\lib\ojdbc14.jar`
`set JAVAHOMEBIN=<my-java-bin>`	`<my-java-bin>`	The proper Java Runtime bin directory. For example, `C:\bea\jdk141_03\bin`

`set URL=jdbc:oracle:thin:@<my-server>:<my-port>:<my-sid>`	`<my-server>:`
	`<my-port>:`
	`<my-sid>`
		URL (server name and port) of the Oracle server and the Oracle SID of the Oracle database (SCPO or Transportation) to access the Oracle database.
		For example,
		`jdbc:oracle:thin:@psdem03:1521:MSAM`

`set USER=<my-manu-access-ID>`	`<my-manu-access-ID>`	ID of the Manugistics database owner
`set PASSWORD=<my-manu-access-password>`	`<my-manu-access-password>`	Password for the Manugistics ID

Table 2-1 Table of Parameters for Manugistics Schema Files

Statement	Replace	With
`set SCHEMA_OWNER=<my-manu-schema-owner>`	`<my-manu-schema-owner>`	The owner of the application tables
`SET OUTPATH="<my-schema-repository-directory>"`	`"<my-schema-repository-directory>"`	Path where schemas will be stored. For more informa on how to set this path, see Naming a Schema Reposi If the path has spaces, it must be enclosed in double qu ("").

3. Save the file.

Running the Schema-Generation Utilities

You must create service and event schemas before you configure services and events in the WebLogic Integration environment. You must run the utilities for each service and for each event. The utilities place the schemas in the directory you specified when editing the scripts. To learn more about editing these scripts, see Editing the Schema-Generation Utility Script Files.

To run the utilities:

1. Open a command window.
2. Navigate to the directory to which you extracted the manugistics schema script and the `ibi-edaqm.jar` file.

   For more information on these files, see Extracting the Schema-Generation Utilities.

3. From the command prompt, run the command as follows for the appropriate database and service or event:

   Note: To issue the commands listed below on UNIX systems, be sure to include `./` before the command; for example,

   ```
   ./manu_trns_schema.sh
   ```

   In addition, you might have to change UNIX permissions to make the script executable.

   - For SCPO batch services or events the command is the following:

     ```
     manu_scpo_schema [SERVICE | EVENT] Desktop userview
     ```

Here, specify whether you are running the utility for a service or event; *`Desktop`* is the name of the Manugistics desktop, and *`userview`* is the particular Manugistics userview.

For example, if you are running the utility to generate a service schema for the Demand Planning desktop from the History userview, type the following at the command prompt:

```
manu_scpo_schema SERVICE DEMANDPLANNING HISTORY
```

- For SCPO SQL events the command is the following:

  ```
  manu_scpo_schema TABLE TableName
  ```

  Here, *`TableName`* is the name of the RDBMS table on which you are listening.

- For Transportation batch services or events not using the Generic Interface the command is the following:

  ```
  manu_trns_schema [SERVICE | EVENT] InterfaceName
  ```

  Here, *`InterfaceName`* is the name of the Transportation interface with which you are communicating.

- For Transportation batch services or events using the Generic Interface the command is the following:

  ```
  manu_trns_schema [SERVICE | EVENT] GENERIC TableName
  ```

  Here, *`TableName`* is the name of the Transportation database table with which the Generic Interface is working.

Listing 2-2 Sample Service Schema Success Message on a Windows System

```
Service Schemas successfully created for User View:
        DESKTOP: DEMANDPLANNING
        USERVIEW: HISTORY
        in Directory:
        C:\Program Files\BEA Systems\BEA Application
Explorer\sessions\default\MGISTICS\psdem03\
```

Next Steps

After you have defined schemas for your events and services, the next step is to create an application view. An application view makes the services and events available to applications. To learn more about application views, see Defining Application Views for Manugistics.

CHAPTER 3

Defining Application Views for Manugistics

An application view is a business-oriented interface to objects and operations within an EIS.

This section presents the following topics:

- How to Use This Document
- Before You Begin
- About Application Views
- About Defining Application Views
- Defining Service Connection Parameters
- Setting Service Properties
- Setting Event Properties
- Defining Event Connection Parameters
- Testing Services
- Testing Events Using a Service
- Testing Events Manually

How to Use This Document

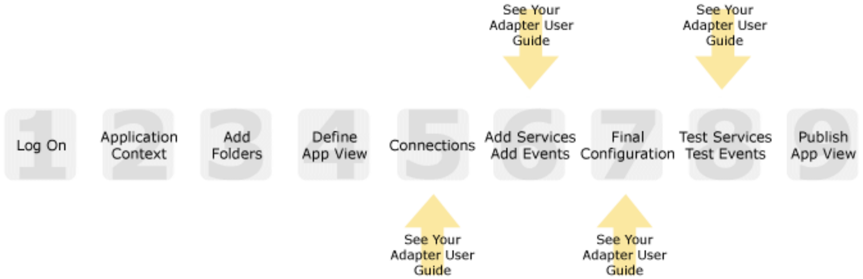
This document is designed to be used in conjunction with *Using the Application Integration Design Console*, available at the following URL:

http://edocs.bea.com/wli/docs81/aiuser/index.html

Using the Application Integration Design Console describes, in detail, the process of defining an application view, which is a key part of making an adapter available to process designers and other users. What *Using the Application Integration Design Console* does *not* cover is the specific information—about connections to your Manugistics system, as well as supported services and events—that you must supply as part of the application view definition. You will find that information in this section.

At each point in *Using the Application Integration Design Console* where you need to refer to this document, you will see a note that directs you to a section in your adapter user guide, with a link to the edocs page for adapters. The following road map illustration shows where you need to refer from *Using the Application Integration Design Console* to this document.

Figure 3-1 Information Interlock with *Using the Application Integration Design Console*


See Your Adapter User Guide
See Your Adapter User Guide
1 Log On
2 Application Context
3 Add Folders
4 Define App View
5 Connections
6 Add Services Add Events
7 Final Configuration
8 Test Services Test Events
9 Publish App View
See Your Adapter User Guide
See Your Adapter User Guide

Before You Begin

Before you define an application view, make sure you have:

- Installed and deployed the adapter according to the instructions in *BEA WebLogic Adapter for Manufistics Installation and Configuration Guide*.
- Determined which business processes need to be supported by the application view. The required business processes determine the types of services and events you include in your application views. Therefore, you must gather information about the application's business requirements from the business analyst. Once you determine the necessary business

processes, you can define and test the appropriate services and events. For more information, see Getting Started With the Adapter for Manugistics.

- Gathered the connection information for your Manugistics system.
- Created interface tables if you are using Manugistics SQL Events. To learn more about creating these interface tables, see Creating Interface Tables for Manugistics SQL Events.
- Configured the Manuba batch process. To learn more about configuring the Manuba batch process, see How the Adapter Works With the Manuba Batch Process.

Creating Interface Tables for Manugistics SQL Events

The BEA WebLogic Adapter for Manugistics receives events from your Manugistics application in two ways:

- Through the standard Manugistics Supply Chain Planning and Optimization (SCPO) and Transportation application export facilities, also known as batch events.
- By listening for changes on your Manugistics database(s) (SQL events).

These two approaches are complementary in nature. The export facilities typically combine data from multiple RDBMS sources and create output structures that reflect the underlying business rules. In addition, the application export facilities allow you to choose the time of the event delivery and provide filtering capabilities. By contrast, SQL events deliver information in near real time and provide information from one RDBMS table only, though you can configure SQL listeners for different tables.

The interface tables described in this section are required only if you are using SQL events. SQL events rely on interface tables that temporarily hold data from the base application tables and database triggers that populate the interface tables with that data. One trigger and one interface table are required for each underlying database table on which you want to listen for changes. You can create the table and trigger by using one of the SQL scripts included in the `BEA_MGISTICS_8_1.manifest.zip` file for Windows systems and `BEA_MGISTICS_8_1.manifest.tar` for UNIX systems, both of which are included in the adapter EAR file.

Two script files are included in the `BEA_MGISTICS_8_1.manifest.zip` and `BEA_MGISTICS_8_1.manifest.tar` files:

- `create_trigger01.sql`

  This script creates the DDL (data definition language) that is used to create the database table and trigger and displays the output in the standard output window.

- `create_trigger02.sql`

  This script creates the DDL for the database table and trigger and immediately runs the DDL to create them.

Caution: The script files must be extracted from the `BEA_MGISTICS_8_1.manifest.zip` or `BEA_MGISTICS_8_1.manifest.tar` file and placed on the Oracle database server that contains the table(s) on which you want to listen for events. The scripts function for any Manugistics Supply Chain Planning and Optimization or Transportation database. For information on running the script files, see Running the Oracle Script files.

Running the Oracle Script files

The script files must be run against the Manugistics Oracle database tables on which you want to listen for events. The script files create an interface table and add a trigger to the Manugistics table specified. You should consult with your Oracle DBA before running the scripts.

To run the scripts:

1. Locate the `BEA_MGISTICS_8_1.manifest.zip` (for Windows systems) or `BEA_MGISTICS_8_1.manifest.tar` file (for UNIX systems) within the `BEA_MGISTICS_8_1.ear` file.

2. Extract the necessary files:

   On Windows: Using WinZip or another utility, extract the `create_trigger01.sql` and `create_trigger02.sql` files from the `BEA_MGISTICS_8_1.manifest.zip` file to the DBS directory on your Oracle installation (for example, *`orahome`*\dbs).

   On UNIX: Issue the following command: `tar -xvf BEA_MGISTICS_8_1.manifest.tar`

   Note: This UNIX command extracts all the files. Be sure to move the `create_trigger01.sql` and `create_trigger02.sql` files into the DBS directory on your Oracle installation (for example, *`orahome`*/dbs)

   Here, *`orahome`* is the base installation directory for your Oracle installation.

3. Edit `create_trigger02.sql` to reflect your Oracle installation directory.

   Change the following line so that it reflects your Oracle installation:

   ```
   DEFINE runfile = ‘C:\oracle\ora817\dbs\create_trigger_t001.sql
   ```

4. Open SQL Plus, a standard Oracle utility and log in to the SCPO database.

5. Run either `create_trigger01.sql` or `create_trigger02.sql` from the Oracle utility against the table(s) on which you want to listen. Both scripts take two parameters, as follows:

   ```
   @create_trigger02 dbowner table
   ```

   Here, *dbowner* is an ID that owns the SCPO database objects, and *table* is the name of table on which you want to listen for events.

   For example, to create a table and trigger to listen to the Manugistics FCST table for a database owner ID of STSC, the command is as follows:

   ```
   @create_trigger02 STSC FCST
   ```

   The `create_trigger01.sql` file displays the DDL in the SQL Plus window. The `create_trigger02.sql` file actually creates the table and the database trigger. For more information on the two files, see "Creating Interface Tables for Manugistics SQL Events" on page 3-3.

   Note: If you chose to run `create_trigger01.sql`, you can modify the DDL generated to meet your site standards or use the DDL as is. Once you are satisfied that the DDL is correct, you can copy and paste it into SQL Plus and run it. You can also run it in any other standard Oracle utility.

How the Adapter Works With the Manuba Batch Process

The BEA WebLogic Adapter for Manugistics relies on the Manugistics Manuba batch process to import data to and export data from the Universal Data Model (UDM) or Manugistics database. Using the Manuba batch process ensures the integrity of the data. Before you can use the Manuba batch process to access the UDM or Manugistics database, you must create a configuration file for the batch process. To learn more about creating Manugistics configuration files, see your Manugistics documentation.

The configuration file contains information such as the userview. In VIEWpoint, userviews are objects that display data and allow you to interact with the system. Any time you work with data in a database table or in a what-if situation, you are working with a userview. In addition, the configuration file contains information about data selection and process.

For example, a typical export process is as follows:

```
manuba fcst_export.lst fcst_export.log
```

The configuration file `fcst_export.lst` has the following components:

```
DESKTOP DEMANDPLANNING
SELECTION FCST_TYPE_1
USERVIEW FCST_EXPORT.
```

The userview FCST_EXPORT is in the DEMANDPLANNING desktop. Use the FCST_TYPE_1 dataselection. The Manuba batch process uses this information to carry out the import of data to the Manugistics SCPO database.

About Application Views

An application view defines:

- Connection information for the EIS, including login information, connection settings, and so on.
- Service invocations, including the information the EIS requires for this request, as well as the request and response schemas associated with the service.
- Event notifications, including the information the EIS publishes and the event schema for inbound messages.

Typically, an application view is configured for a single business purpose and contains only the services and events required for that purpose. An EIS might have multiple application views, each defined for a different purpose.

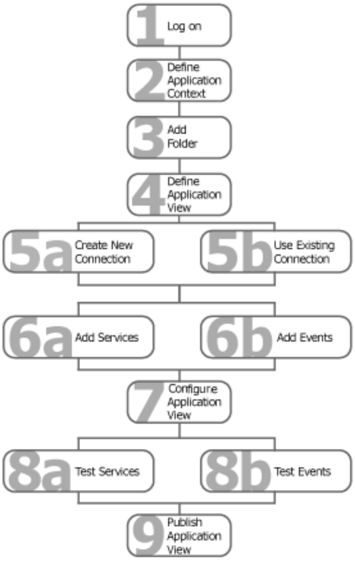
About Defining Application Views

Defining an application view is a multi-step process described in *Using the Application Integration Design Console*, available at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

The information you enter depends on the requirements of your business process and your EIS system configuration. Figure 3-2 summarizes the procedure for defining and configuring an application view.

Figure 3-2 Process for Defining and Configuring an Application View


1 Log on
2 Define Application Context
3 Add Folder
4 Define Application View
5a Create New Connection
5b Use Existing Connection
6a Add Services
6b Add Events
7 Configure Application View
8a Test Services
8b Test Events
9 Publish Application View

To define an application view:

1. Log on to the WebLogic Integration Application View Console.
2. Define the application context by selecting an existing application or specifying a new application name and root directory.

   This application will be using the events and services you define in your application view. The application view works within the context of this application.
3. Add folders as required to help you organize application views.
4. Define a new application view for your adapter.
5. Add a new connection service or select an existing one.

   If you are adding a new connection service, see "Defining Service Connection Parameters" on page 3-8 for details about Manugistics requirements.
6. Add the events and services for this application view.

   See the following sections for details about Manugistics requirements:

   – "Setting Service Properties" on page 3-9

- “Setting Event Properties” on page 3-14

7. Perform final configuration tasks.

   If you are adding an event connection, see “Defining Event Connection Parameters” on page 3-21 for details about Manugistics requirements.

8. Test all services and events to make sure they can properly interact with the target Manugistics system.

   See the following sections for details about Manugistics requirements:

   - “Testing Services” on page 3-22
   - “Testing Events Using a Service” on page 3-23
   - “Testing Events Manually” on page 3-24

9. Publish the application view to the target WebLogic Workshop application.

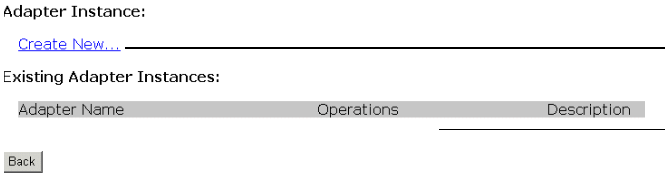
   This is the application you specified in step 2. Publishing the application view allows workflow developers within the target application to interact with the newly published application view using an Application View control.

Defining Service Connection Parameters

This information applies to “Step 5A, Create a New Browsing Connection” in *Using the Application Integration Design Console*, at the following URL:

http://edocs.bea.com/wli/docs81/aiuser/index.html

The Select Browsing Connection page allows you to choose the type of connection factory to associate with the application view. You can select a connection factory within an existing instance of the adapter or create a connection factory within a new adapter instance.



Click to create a new connection factory

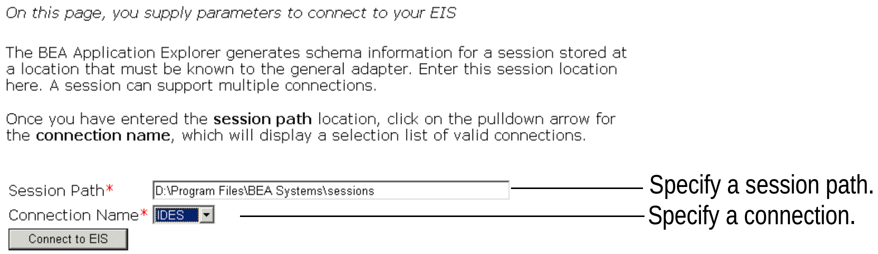
Existing connection factories will be here

After you enter a connection name and description, you use the Configure Connection Parameters page to specify connection parameters for a connection factory.

To create a new browsing connection:

1. In the Create New Browsing Connections page, enter a connection name and description as described in *Using the Application Integration Design Console*.

   The Configure Connection Parameters page appears to allow you to configure the newly created connection factory within the new adapter instance.



On this page, you supply parameters to connect to your EIS

The BEA Application Explorer generates schema information for a session stored at a location that must be known to the general adapter. Enter this session location here. A session can support multiple connections.

Once you have entered the **session path** location, click on the pulldown arrow for the **connection name**, which will display a selection list of valid connections.

Session Path* D:\Program Files\BEA Systems\sessions

Connection Name* IDES

Specify a session path.

Specify a connection.

Note: A red asterisk (*) indicates that a field is required.

2. Specify a session path and connection name.

   This information enables the application view to interact with the target Manugistics system. You need enter this information only once per application view. To learn more about this, see Defining Schemas.

3. Click Connect to EIS.

   You return to the Create New Browsing Connections, where you can specify connection pool parameters and logging levels. For more information, see *Using the Application Integration Design Console* at the following URL:

   `http://edocs.bea.com/wli/docs81/aiuser/index.html`

Setting Service Properties

This information applies to "Step 6A, Add a Service to an Application View" in *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

Adapter for Manugistics uses services to make requests of the Manugistics system. A service consists of both a request and a response. The Adapter for Manugistics supports the following services:

- Manu SCPO Service
- Manu TRNS Service

Manu SCPO Service

This information applies to “Step 6A, Add a Service to an Application View” in *Using the Application Integration Design Console*, at the following URL:

http://edocs.bea.com/wli/docs81/aiuser/index.html

An SCPO service sends a request to the Manugistics Supply Chain application using the Manuba batch process. Manugistics processes the request and sends back a response. To learn more about configuring the Manuba batch process, see your Manugistics documentation.

Note: In order for your Manuba batch file (`manuba.bat`) to work properly, ensure the following environment variables are set:

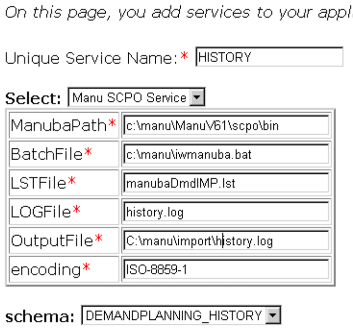
```
SET ORACLE_SID=MSAM
SET ORALOGON=username/password
SET MANUDEFAULTS=scpo_init_file_name
```

These environment variables are typically created and set at the time of your Manugistics SCPO installation.

To configure a Manu SCPO Service:

1. Enter a unique service name that describes the function the service performs.
2. Select Manu SCPO Service from the Select list.

   The Add Services page displays the fields required for this service type.



On this page, you add services to your application view.

Unique Service Name:* HISTORY

Select: Manu SCPO Service

ManubaPath*	c:\manu\ManuV61\scpo\bin
BatchFile*	c:\manu\iwmanuba.bat
LSTFile*	manubaDmdIMP.lst
LOGFile*	history.log
OutputFile*	C:\manu\import\history.log
encoding*	ISO-8859-1

schema: DEMANDPLANNING_HISTORY

Note: A red asterisk (*) indicates that a field is required.

3. Enter the following information:

Table 3-1 Manu SCPO Service Parameters

Parameter	Description
ManubaPath	Path to the Manuba process. For example, `c:\manu\manuv61\scpo\bin`
BatchFile	The full path and file name of the batch process that sets the Manuba environment and invokes it.
LSTFile	The configuration file required by Manuba. For example, `manubaDmdIMP.lst`
LOGFile	The name of the output log required by Manuba.
OutputFile	The name of the file containing the formatted data. This file is configured in the Manugistics ViewPoint client for the import process.
encoding	Character encoding of the data. This value defaults to UTF-8.

4. See "Common Service and Event Settings" on page 3-13 for information about selecting a schema and configuring logging and tracing.

Manu TRNS Service

1 2 3 4 5 6 7 8 9

This information applies to "Step 6A, Add a Service to an Application View" in *Using the Application Integration Design Console*, at the following URL:

http://edocs.bea.com/wli/docs81/aiuser/index.html

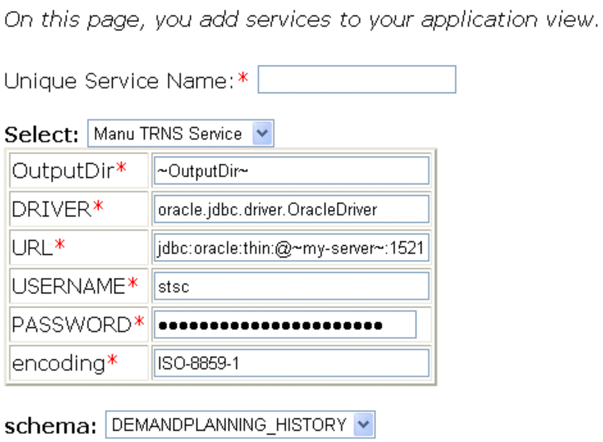
A TRNS service sends a request to the Manugistics Transportation application using the Manuba batch process. Manugistics responds to the request and sends back a response.

To configure a Manu TRNS Service:

1. Enter a unique service name that describes the function the service performs.

2. Select Manu TRNS Service from the Select list.

   The Add Services page displays the fields required for this service type.



On this page, you add services to your application view.

Unique Service Name:*

Select: Manu TRNS Service

OutputDir*	~OutputDir~
DRIVER*	oracle.jdbc.driver.OracleDriver
URL*	jdbc:oracle:thin:@~my-server~:1521
USERNAME*	stsc
PASSWORD*	••••••••••••••••••••••
encoding*	ISO-8859-1

schema: DEMANDPLANNING_HISTORY

Note: A red asterisk (*) indicates that a field is required.

3. Enter the following information:

Table 3-2 Manu TRNS Service Parameters

Parameter	Description
OutputDir	The full path and file name of the export file. **Note:** This must exactly match the path name of the TRP_IN directory which is configured for your Transportation application. To learn more about this, see your Manugistics documentation.
Driver	Vendor-specific JDBC driver for access to the database. This parameter requires a fully-qualified name.
URL	The URL (or JDBC URL) is a platform-independent way of addressing the database. A database/JDBC URL has the following form: `jdbc:[subprotocol]:[node]/[databaseName]`
Username	Valid user name for access to the database
Password	Valid password associated with the user name for access to the database
encoding	Character encoding of the data. This value defaults to UTF-8.

4. See “Common Service and Event Settings” on page 3-13 for information about selecting a schema and configuring logging and tracing.

Common Service and Event Settings

This information applies to “Step 6A, Add a Service to an Application View” in *Using the Application Integration Design Console*, at the following URL:

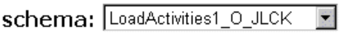
http://edocs.bea.com/wli/docs81/aiuser/index.html

You select a schema and select tracing options the same way for all services and events.

To set common service and event settings:

1. In the Schema list, select the schema you want to use with this service or event.

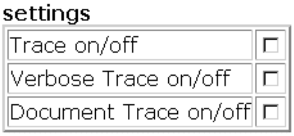
   For more information, see Chapter 2, “Generating Schemas for Manugistics Integration Objects.”

   schema: LoadActivities1_O_JLCK

2. Configure tracing for this service or event, as follows:

   Tracing displays runtime information in the console. You set the type and amount of information you wish to capture as part of the final configuration tasks. This is described in detail in *Using the Application Integration Design Console*.

   settings

Trace on/off	□
Verbose Trace on/off	□
Document Trace on/off	□

   a. Select the Trace on/off check box to enable tracing for this service or event. Trace information appears in the runtime console.

   b. Select the Verbose Trace on/off check box to enable more detailed tracing for this service or event. Trace information appears in the runtime console.

   c. Select the Document Trace on/off check box to enable recording of request, response, and event documents for deeper troubleshooting.

3. Click Add to add the service or event.

   For more information about the next step, see *Using the Application Integration Design Console* at the following URL:

   http://edocs.bea.com/wli/docs81/aiuser/index.html

Setting Event Properties

This information applies to “Step 6B, Add an Event to an Application View” in *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

An event defines how your application responds to events generated by Manugistics. The Adapter for Manugistics supports the following events:

- Manu SQL Event
- Manu SCPO Batch Event
- Manu TRNS Batch Event

Manu SQL Event

This information applies to “Step 6B, Add an Event to an Application View” in *Using the Application Integration Design Console*, at the following URL:

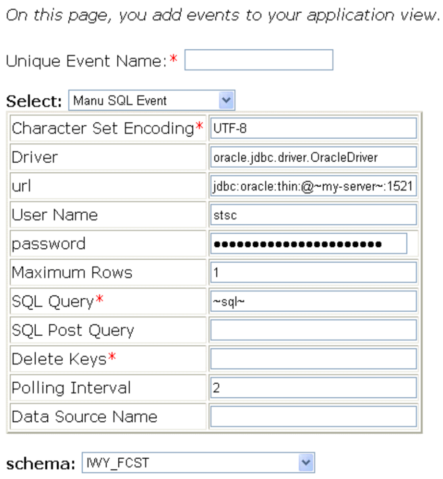
`http://edocs.bea.com/wli/docs81/aiuser/index.html`

In a SQL Event, the adapter picks up an SQL event XML file from Manugistics and passes it to an event variable that is set in a business process.

To configure a Manu SQL Event:

1. Enter a unique event name that describes the function the event performs.
2. Select Manu SQL Event from the Select list.

   The Add Events page displays the fields required for this event type.



On this page, you add events to your application view.

Unique Event Name:*

Select: Manu SQL Event

Character Set Encoding*	UTF-8
Driver	oracle.jdbc.driver.OracleDriver
url	jdbc:oracle:thin:@~my-server~:1521
User Name	stsc
password	••••••••••••••••••••••
Maximum Rows	1
SQL Query*	~sql~
SQL Post Query	
Delete Keys*	
Polling Interval	2
Data Source Name	

schema: IWY_FCST

Note: A red asterisk (*) indicates that a field is required.

3. Enter the following information:

Table 3-3 Manu SQL Event Parameters

Parameter	Description
Character Set encoding	Encoding of the data being read
Driver	Vendor-specific JDBC driver for access to the database. This parameter requires a fully-qualified name.
url	The URL (or JDBC URL) is a platform-independent way of addressing the database. A database/JDBC URL has the following form: `jdbc:[subprotocol]:[node]/[databaseName]`
User Name	Valid user name for access to the database
password	Valid password associated with the user name for access to the database

Table 3-3 Manu SQL Event Parameters (Continued)

Parameter	Description
Maximum Rows	Number of data rows to be retrieved from the database table in a single operation. For example, if you specified five, then up to five rows are read and processed in a single operation. In most circumstances, you should not allow this parameter to exceed the number of parallel threads available for execution. The number of events created in a single polling interval is dependent on the setting for Maximum Rows, and the number of new rows added to the database since the last time the database was polled. For example, if Maximum Rows is set to 5, and 23 new rows are found to have been added when the database is polled, four events containing five rows and one event containing three rows are created.
SQL Query	Enter a SQL query for the table from which you want data. All tablenames are prepended with IWY_. For example, to get FCST information, select * from IWY_FCST These tables are created by the create_trigger_*.sql scripts.
SQL Post Query	SQL Query that is executed after the initial query request. If this parameter is not configured, the following command is executed: DELETE *field1,field2...* from *table_name* Do not configure this parameter if the RDBMS event adapter exit is configured. Two types of operators are available: ?fieldname and ^fieldname. ■ The ?fieldname will evaluate at run time to ?fieldname= value. ■ The ^fieldname will evaluate at run time to value. A SQL Post query using the ? can be used in an update statement as follows: update *tablename* where ?fieldname. For example, update stock_prices_temp where ?RIC. A SQL Post Query using the ^ can be used in an insert statement as follows: Insert into *tablename values* (^fieldname1, ^fieldname2, ^fieldname3). For example, Insert into stock_prices_temp values (^RIC, ^PRICE, ^UPDATED).

Table 3-3 Manu SQL Event Parameters (Continued)

Parameter	Description
Delete Keys	Comma separated list of keys used in the DELETE statement. A delete operates on keys, so you should enter the table's key columns in this parameter.
Polling Interval	Interval in seconds at which the database is monitored for new rows. If this parameter is not configured, the default value is two seconds.
Data Source Name	The Data Source JNDI name for the JDBC connection pool to use for connecting to the RDBMS system. If a value is present, the adapter will use the connection pool to connect to the RDBMS. If no value is specified, connection will use the Driver, URL, User Id, and Password specified for this event. **Note:** You can use either the URL, Driver, and Password, OR the data source name to connect to the Manugistics database.

4. See "Common Service and Event Settings" on page 3-13 for information about selecting a schema and configuring logging and tracing.

Manu SCPO Batch Event

This information applies to "Step 6B, Add an Event to an Application View" in *Using the Application Integration Design Console*, at the following URL:

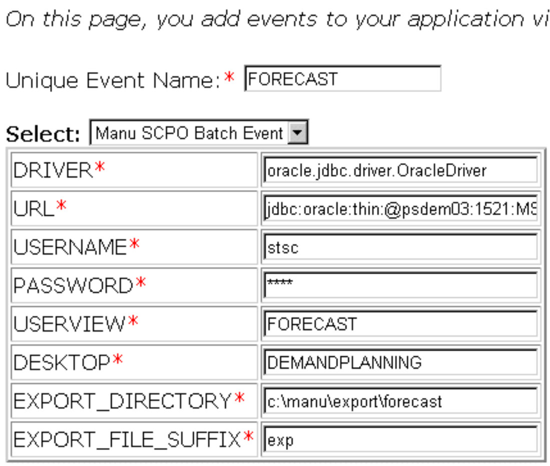
http://edocs.bea.com/wli/docs81/aiuser/index.html

In an SCPO batch event, the adapter picks up an event XML file and passes it to an event variable that is set in a business process.

To configure a Manu SCPO Batch Event:

1. Enter a unique event name that describes the function the event performs.
2. Select Manu SCPO Batch Event from the Select list.

   The Add Events page displays the fields required for this event type.



On this page, you add events to your application view.

Unique Event Name:* FORECAST

Select: Manu SCPO Batch Event

DRIVER*	oracle.jdbc.driver.OracleDriver
URL*	jdbc:oracle:thin:@psdem03:1521:MS
USERNAME*	stsc
PASSWORD*	****
USERVIEW*	FORECAST
DESKTOP*	DEMANDPLANNING
EXPORT_DIRECTORY*	c:\manu\export\forecast
EXPORT_FILE_SUFFIX*	exp

Note: A red asterisk (*) indicates that a field is required.

3. Enter the following information:

Table 3-4 Manu SCPO Batch Event Parameters

Parameter	Description
Driver	Vendor-specific JDBC driver for access to the database. This parameter requires a fully-qualified name.
URL	The URL (or JDBC URL) is a platform-independent way of addressing the database. A database/JDBC URL has the following form: `jdbc:[subprotocol]:[node]/[databaseName]`
Username	Valid user name for access to the database
Password	Valid password associated with the user name for access to the database
UserView	The name of the Manugistics userview you are listening on for event data.
Desktop	The name of the Manugistics desktop for the specified userview
EXPORT_DIRECTORY	The full path of the directory for Manugistics batch export file for the specified userview. **Note:** This must exactly match the path as entered in the Manugistics export facility for the specified userview.

Table 3-4 Manu SCPO Batch Event Parameters (Continued)

Parameter	Description
EXPORT_FILE_SUFFIX	Suffix of the file being deposited in the export directory

4. See "Common Service and Event Settings" on page 3-13 for information about selecting a schema and configuring logging and tracing.

Manu TRNS Batch Event

This information applies to "Step 6B, Add an Event to an Application View" in *Using the Application Integration Design Console*, at the following URL:

http://edocs.bea.com/wli/docs81/aiuser/index.html

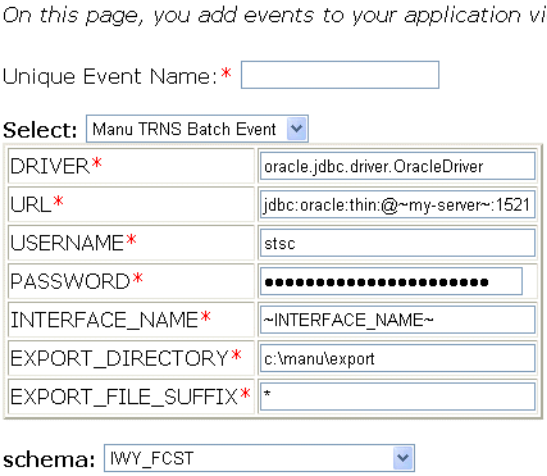
In a TRNS batch event, the adapter picks up an event XML file and passes it to an event variable that is set in a business process.

To configure a Manu TRNS Batch Event:

1. Enter a unique event name that describes the function the event performs.

2. Select Manu TRNS Batch Event from the Select list.

   The Add Events page displays the fields required for this event type.



On this page, you add events to your application view.

Unique Event Name:*

Select: Manu TRNS Batch Event

DRIVER*	oracle.jdbc.driver.OracleDriver
URL*	jdbc:oracle:thin:@~my-server~:1521
USERNAME*	stsc
PASSWORD*	●●●●●●●●●●●●●●●●●●●●●●●
INTERFACE_NAME*	~INTERFACE_NAME~
EXPORT_DIRECTORY*	c:\manu\export
EXPORT_FILE_SUFFIX*	*

schema: IWY_FCST

Note: A red asterisk (*) indicates that a field is required.

3. Enter the following information:

Table 3-5 Manu TRNS Batch Event Parameters

Parameter	Description
Driver	Vendor-specific JDBC driver for access to the database. This parameter requires a fully-qualified name.
URL	The URL (or JDBC URL) is a platform-independent way of addressing the database. A database/JDBC URL has the following form: `jdbc:[subprotocol]:[node]/[databaseName]`
Username	Valid user name for access to the database
Password	Valid password associated with the user name for access to the database
INTERFACE_NAME	Name of the Manugistics Transportation interface that creates the file found in the EXPORT_DIRECTORY.
EXPORT_DIRECTORY	The full path of the directory for the Manugistics Transportation batch export file. This takes the form: *`directory_name\file_pattern`* (Windows) or *`directory_name/file_pattern`* (UNIX). Here, *`directory_name`* is the location in which the Transportation output interface processor, identified as `TRP_OUT` in the Manugistics documentation, places the output files, and *`file_pattern`* is `*rt` for ROUTED_ORDER and `table_name*` for Generic Output Interfaces. For example, if `TRP_OUT` is `C:\Manu\ManuV622\tserver\out` and you are expecting data using the generic output for the `MARINE_SCHED` table, the `EXPORT_DIRECTORY` is `C:\Manu\ManuV622\tserver\out\marine_sched*`.
EXPORT_FILE_SUFFIX	Suffix of the file being deposited in the export directory. • For generic interface events, use `Dat` • For routed order interface events, use `*`

4. See "Common Service and Event Settings" on page 3-13 for information about selecting a schema and configuring logging and tracing.

Defining Event Connection Parameters

This information applies to "Step 7, Perform Final Configuration Tasks" in *Using the Application Integration Design Console*, at the following URL:

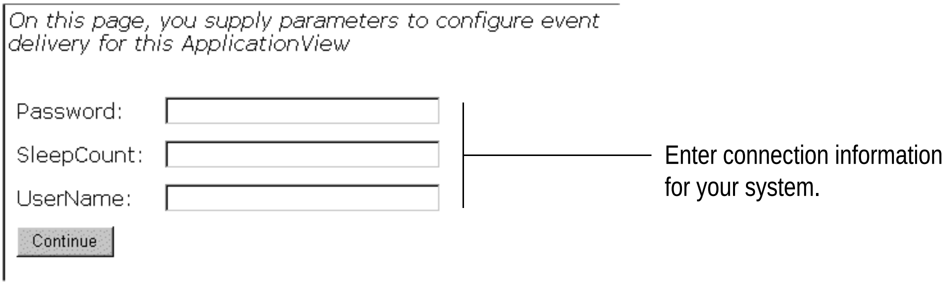
`http://edocs.bea.com/wli/docs81/aiuser/index.html`

Once you have finished adding services and events and have saved your application view, you must perform some final configuration tasks, including configuring event delivery connections, before testing the services and events. You perform these configuration tasks from the Final Configuration and Testing page.

To define event connection parameters:

1. In Connections area on the Application View Administration page, click Select/Edit.
2. In the Event Connection area, click Event to edit the default event connection.

   The Configure Event Delivery Parameters page appears.



On this page, you supply parameters to configure event delivery for this ApplicationView

Enter connection information for your system.

Note: A red asterisk (*) indicates that a field is required.

3. Enter the following information:

Table 3-6 Event Connection Parameters

Parameter	Description
username	Your WebLogic Server Administration Console user name, defined in the `startWebLogic` script
password	The password for your WebLogic Server Administration Console user name
SleepCount	The number of seconds the adapter will wait between polling for events

The event delivery parameters you enter on this page enable connection to your Manugistics system and are used when generating events. The parameters are specific to the associated adapter and are defined in the `wli-ra.xml` file within the base adapter.

4. Click Save to save your event delivery parameter settings. Click Continue to return to the Edit Event Adapter page, and then click Back to return to the Final Configuration and Testing page.

   The Edit Event Adapter page allows you to define event parameters and configure the information that will be logged for the connection factory. Select one of the following settings for the log:

   – Log errors and audit messages

   – Log warnings, errors, and audit messages

   – Log informational, warning, error, and audit messages

   – Log all messages

Note: For maximum tracing, select Log all Messages. This is the recommended setting to use when you are collecting debugging information for BEA support.

The table that follows describes the type of information that each logging message contains.

Table 3-7 Logging message categories

This type of message	Contains
Audit	Extremely important information related to the business processing performed by an adapter.
Error	Information about an error that has occurred in the adapter, which may affect system stability.
Warning	Information about a suspicious situation that has occurred. Although this is not an error, it could have an impact on adapter operation.
Information	Information about normal adapter operations.

Testing Services

This information applies to “Step 8A, Test an Application View’s Services” in *Using the Application Integration Design Console*, at the following URL:

http://edocs.bea.com/wli/docs81/aiuser/index.html

The purpose of testing an application view service is to evaluate whether that service interacts properly with the target Manugistics system. When you test a service, you supply any inputs required to start the service. For the Adapter for Manugistics, the input is in the form of a valid XML string that acts as input for the service.

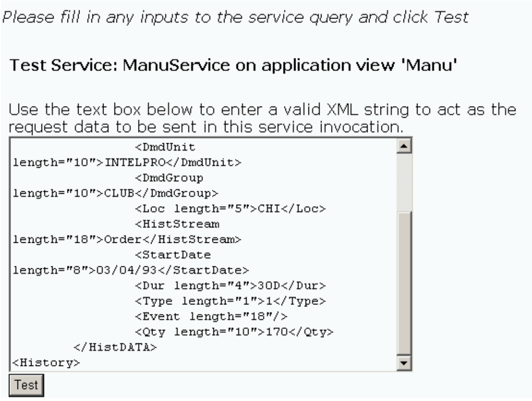
Note: You can test an application view only if it is deployed and only if it contains at least one event or service.

To test a service:

1. In the Application View Administration page, click the Test link beside the service to be tested.

   The Test Services page appears.

2. In the Test Service window, copy the appropriate XML strings for your account.



Please fill in any inputs to the service query and click Test

Test Service: ManuService on application view 'Manu'

Use the text box below to enter a valid XML string to act as the request data to be sent in this service invocation.

```
                <DmdUnit
length="10">INTELPRO</DmdUnit>
                <DmdGroup
length="10">CLUB</DmdGroup>
                <Loc length="5">CHI</Loc>
                <HistStream
length="18">Order</HistStream>
                <StartDate
length="8">03/04/93</StartDate>
                <Dur length="4">30D</Dur>
                <Type length="1">1</Type>
                <Event length="18"/>
                <Qty length="10">170</Qty>
        </HistDATA>
<History>
```

3. Click Test.

   The results appear in the Test Results window.

Testing Events Using a Service

This information applies to “Step 8B, Test an Application View’s Events” in *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

The purpose of testing an application view event is to make sure that the adapter correctly handles events generated by Manugistics. When you test an event, you can trigger the event using a service or manually.

Note: You can test an application view only if it is deployed and only if it contains at least one event or service.

To test an event:

1. In the Application View Administration page, click the Test link beside the event to be tested.

   The Test Events page appears.

2. Click Service and select a service that triggers the event you are testing.

3. In the Time field, enter a reasonable period of time to wait, specified in milliseconds, before the test times out (One second = 1000 milliseconds. One minute = 60,000 milliseconds.).

4. Click Test and enter the XML string needed to trigger the service.

   The service is executed.

   – If the test succeeds, the Test Result page appears, showing the event document, the service input document, and the service output document.

   – If the test fails, the Test Result page displays only a Timed Out message.

Testing Events Manually

This information applies to “Step 8B, Test an Application View’s Events” in *Using the Application Integration Design Console*, at the following URL:

`http://edocs.bea.com/wli/docs81/aiuser/index.html`

To test an event manually:

1. In the Application View Administration page, click the Test link beside the event to be tested.

2. In the Time field, enter a reasonable period of time to wait, specified in milliseconds, before the test times out (One second = 1000 milliseconds. One minute = 60,000 milliseconds.).

3. Click Test. The test waits for an event to trigger it.

4. Using the triggering Manugistics application, perform an action that generates the event.

- If the test succeeds, the Test Result page appears. This page displays the event document from the application.
- If the test fails or takes too long, the Test Result page appears, showing a Timed Out message.

Index