SeeBeyond ICAN Suite

RFID eWay Intelligent Adapter User's Guide

Release 5.0.1

SEEBEYOND®

SeeBeyond Proprietary and Confidential

The information contained in this document is subject to change and is updated periodically to reflect changes to the applicable software. Although every effort has been made to ensure the accuracy of this document, SeeBeyond Technology Corporation (SeeBeyond) assumes no responsibility for any errors that may appear herein. The software described in this document is furnished under a License Agreement and may be used or copied only in accordance with the terms of such License Agreement. Printing, copying, or reproducing this document in any fashion is prohibited except in accordance with the License Agreement. The contents of this document are designated as being confidential and proprietary; are considered to be trade secrets of SeeBeyond; and may be used only in accordance with the License Agreement, as protected and enforceable by law. SeeBeyond assumes no responsibility for the use or reliability of its software on platforms that are not supported by SeeBeyond.

SeeBeyond, e*Gate, e*Way, and e*Xchange are the registered trademarks of SeeBeyond Technology Corporation in the United States and/or select foreign countries. The SeeBeyond logo, SeeBeyond Integrated Composite Application Network Suite, eGate, eWay, eInsight, eVision, eXchange, eView, eIndex, eTL, ePortal, eBAM, and e*Insight are trademarks of SeeBeyond Technology Corporation. The absence of a trademark from this list does not constitute a waiver of SeeBeyond Technology Corporation's intellectual property rights concerning that trademark. This document may contain references to other company, brand, and product names. These company, brand, and product names are used herein for identification purposes only and may be the trademarks of their respective owners.

© 2005 by SeeBeyond Technology Corporation. All Rights Reserved. This work is protected as an unpublished work under the copyright laws.

This work is confidential and proprietary information of SeeBeyond and must be maintained in strict confidence.

Version 20050502131010.

Contents

Chapter 1

Introducing the RFID eWay 6

About RFID 6

About the RFID eWay 7
Key Features of the RFID eWay 7
RFID eWay Modes of Operation 8

What's New in This Release 8

About This Document 8
Organization of Information in This Book 8
Scope 9
Intended Audience 9
Document Conventions 9
Screenshots 10

Related Documents 10

SeeBeyond Web Site 10

SeeBeyond Documentation Feedback 11

Chapter 2

Installing the RFID eWay 12

Supported Operating Systems 12

System Requirements 12
Prerequisite ICAN Modules 13
JMS Requirements 13

External System Requirements 13

Installing the eWay Product Files 14

Configuring Third-Party Event Managers 14
Configuring the Sun RFID Event Manager 14
Configuring the Sun Event Manager with RFID readers 16
Running the Sun RFID Event Manager 19
Configuring the ConnecTerra Event Manager 20
Configuring the ConnecTerra Event Manager with RFID Readers 21
Running the ConnecTerra Event Manager using HTTP 22
Running Your ConnecTerra Event Manager Using JMS 23

Using the JMS binding utility 23

Chapter 3

Configuring the RFID eWay 26

Creating RFID eWay Projects 26
Using the Reports and EDI Options 27

Sun RFID Projects 28
Using the SunRFIDSuiteJMSOracleTemplate Project 29
Using the SunRFIDSuiteJMSDB2Template Project 37
Using the SunRFIDSuiteJMSSQLServerTemplate Project 40
Using the SunRFIDSuiteJMSTigerLogicTemplate Project 43

ConnecTerra Projects 49
Using the ConnecterraJMSOracleTemplate Project 50
Using the ConnecterraJMSDB2Template Project 53
Using the ConnecterraJMSSQLServerTemplate Project 56
Using the ConnecterraJMSTigerLogicTemplate Project 59
Using the HTTPJMSQueueTemplate Project 65
Using the ConnecterraHTTPOracleTemplate Project 69
Using the ConnecterraHTTPDB2Template Project 72
Using the ConnecterraHTTPSQLServerTemplate Project 75
Using the ConnecterraHTTPTigerLogicTemplate Project 78

Chapter 4

Creating the Observation_Log Database 84

The Observation_Log Database 84
Observation_Log SQL Files 84

Creating the Tables 85

Archiving the Transactions 86

Additional Tables for the Sample EPC IS Application 87

Populating the Tables with Sample Data 88

Chapter 5

Working with the EPC Event Generator 89

What is the EPC Event Generator? 89

Create a New RFID Application 90

Configure EPCGeneratorOracleTemplate 92

Configure EPCEventGenerator 93

Run the Logical Host 94

Run the Event Generator 95
View the Results 97

Chapter 6

Using the RFID eWay Reports 98

The RFID Reports for the Sample EPC IS Application 98
Activating Reports for the Sample EPC IS Application 99
Inventory Report 106
OverStock Report 108
Transit Time Report 110
EPC Trace Report 112
PO Reconciliation by Item Report 115
PO Reconciliation by SSCC Report 117
Using the Worklist Manager Alerts 119

Chapter 7

Using the Single WebService Interface for Sample EPC IS 122

About Web Services Operations 122
Configuring Web Services Operations 122
Running a Sample Web Services Operation 124

Chapter 8

Using the RFID eWay EDI Option 129

Activating EDI for the Sample EPC IS Application 129

Index 131

Chapter 1

Introducing the RFID eWay

Welcome to the *RFID eWay Intelligent Adapter User's Guide*. This document includes information about installing, configuring, and using the SeeBeyond® Integrated Composite Application Network Suite™ (ICAN) RFID eWay Intelligent Adapter, referred to as the RFID eWay throughout this guide.

What's in This Chapter

- **"About RFID" on page 6**
- **"About the RFID eWay" on page 7**
- **"What's New in This Release" on page 8**
- **"About This Document" on page 8**
- **"Related Documents" on page 10**
- **"SeeBeyond Web Site" on page 10**
- **"SeeBeyond Documentation Feedback" on page 11**

1.1 About RFID

Radio frequency identification (RFID) is a technology that uses wireless transmission to send unique product codes from product tags (RFID tags) to a scanner, without any human intervention. RFID technology is similar to barcode scanning, except that RFID uses radio frequencies instead of optics and does not require a person to scan an item's barcode. Unlike a typical barcode though, the information included in the RFID tag also includes a unique serial number.

An RFID system consists of an antenna, a transceiver (or reader), and a transponder (or tag). The antenna uses radio frequency waves to transmit a signal that activates the transponder (for passive tags). When activated, the transponder transmits data back to the reader via the antenna. The data is captured by the RFID system and initiates the logic built into the system. This could be any activity, from simply reporting the movement of an inventory item, to opening a secured door in an office building, authorizing a fuel transaction at the pump, or even accessing a database and performing a complex transaction.

1.2 About the RFID eWay

The RFID eWay works to simplify data being transmitted by RFID systems. RFID systems typically send enormous amounts of data every day. An Event Manager can aggregate this data so that duplicate records are eliminated. The RFID eWay works to persist data being transmitted by event managers that interface directly with the readers. Once the data is persisted, you can use it to populate your EPC Information Service application or other applications. The RFID eWay also provides a sample EPC Information Service application to help you get started using the eWay.

1.2.1 Key Features of the RFID eWay

The key features of the RFID eWay include:

- **Interface with Event Managers:** The RFID eWay receives data from Event Managers using JMS (for Sun RFID and ConnecTerra) or HTTP (for ConnecTerra). After receiving and processing the RFID tag data, the RFID eWay writes the outbound records into an *observation_log* database table. The supported databases are listed in **Table 1 on page 8**.
- **EPC Information Service:** The RFID eWay provides a sample EPC Information Service application so that users do not have to build the EPC IS from scratch. This application addresses two issues brought up by the EPC Business Action Group: “Track and Trace” and “Ship and Receive”. The sample EPC application uses Oracle as the database for storing data. It contains Oracle table schema, sample data, reports generated by eVision, alerts generated by eInsight Worklist Manager, Java Collaborations processing X12 data (Purchase Order and Advanced Shipping Notice), and optional web service (server side) for user-defined reports. The eVision reports provided are:
    - Inventory
    - Overstock
    - Transit Time
    - EPC Trace
    - PO Reconciliation by Item
    - PO Reconciliation by SSCC
- **Event Generator:** The RFID eWay includes an EPC Event Generator. The event generator is a convenient way of creating sample RFID tag records in a lab environment without the inconvenience of manually scanning RFID tags. The event generator is provided for testing and demonstration purposes only.

1.2.2 RFID eWay Modes of Operation

The RFID eWay can accept input data from different sources and send the output data to various destinations. The following table shows which sources and destinations function with each other.

Table 1 RFID eWay Modes of Operation

Vendor	Input Type(s)	Output Type(s)
Sun RFID Event Manager	JMS	▪ Oracle ▪ DB2/UDB ▪ SQL Server ▪ TigerLogic
ConnecTerra	▪ JMS* ▪ HTTP *See note below	▪ JMS* ▪ Oracle ▪ DB2/UDB ▪ SQL Server ▪ TigerLogic *See note below

Note: *JMS is available as* ***either*** *the Input* ***or*** *Output type. You cannot use JMS for both the Input and the Output types in the same RFID Project.*

For more information on creating a new Project using these options, see **Chapter 5**.

1.3 What's New in This Release

- Replacement of JMS Queue with Topic
- Added support for JMS Binding
- Support for TigerLogic XML database
- Inclusion of command line support for controlling EPC Event Generator
- Added support for WebServices interface for sample EPC IS
- Added support for logging data to a backup file

1.4 About This Document

1.4.1 Organization of Information in This Book

This document includes the following chapters:

- **Chapter 1 "Introducing the RFID eWay"**: Provides an overview description of the product as well as high-level information about this document.

- **Chapter 2 "Installing the RFID eWay"**: Describes the system requirements and provides instructions for installing the RFID eWay.
- **Chapter 3 "Configuring the RFID eWay"**: Provides instructions for configuring the eWay to communicate with your RFID system.
- **Chapter 4 "Creating the Observation_Log Database"**: Explains how to create and archive the Observation_Log database for use with the RFID eWay.
- **Chapter 5 "Working with the EPC Event Generator"**: Provides instructions for using the *EPC Event Generator*. The event generator is an RFID simulator provided for you to use during testing.
- **Chapter 6 "Using the RFID eWay Reports"**: Describes the reports included with the RFID eWay.
- **Chapter 7**: **"Using the Single WebService Interface for EPC IS"**: Describes the WebService interface for multiple operations in EPC IS.
- **Chapter 8 "Using the RFID eWay EDI Option"**: Describes how to add EDI to your RFID eWay Projects.

1.4.2 Scope

This document describes the process of installing, configuring, and running the RFID eWay.

This document does not include specific details about the installation and configuration of any prerequisite ICAN products, such as the HTTP eWay, Oracle eWay, etc. For detailed information on these products, refer to those products' user's guides and Javadocs.

1.4.3 Intended Audience

This guide is intended for experienced computer users who have the responsibility of helping to set up and maintain a fully functioning ICAN Suite system. This person must also understand any operating systems on which the ICAN Suite will be installed and must be thoroughly familiar with Windows-style GUI operations.

1.4.4 Document Conventions

The following conventions are observed throughout this document.

Table 2 Document Conventions

Text	Convention	Example
Names of buttons, files, icons, parameters, variables, methods, menus, and objects	**Bold** text	▪ Click **OK** to save and close. ▪ On the **File** menu, select **Exit.** ▪ Select the **logicalhost.exe** file. ▪ Enter the **timeout** value. ▪ Use the **getClassName()** method. ▪ Configure the **Inbound** File eWay.

Table 2 Document Conventions (Continued)

Text	Convention	Example
Command line arguments, code samples	`Fixed` font. Variables are shown in ***bold italic***.	`bootstrap -p` ***`password`***
Hypertext links	**Blue** text	See **Document Conventions** on page 9
Hypertext links for Web addresses (URLs) or email addresses	**Blue underlined** text	**http://www.seebeyond.com** **docfeedback@seebeyond.com**

1.4.5 Screenshots

Depending on what products you have installed, and how they are configured, the screenshots in this document may differ from what you see on your system.

1.5 Related Documents

The following SeeBeyond documents provide additional information about the ICAN product suite:

- *ASC X12 OTD Library User's Guide*
- *DB2 Universal Database eWay Intelligent Adapter User's Guide*
- *eGate Integrator User's Guide*
- *eInsight Business Process Manager User's Guide*
- *eVision Studio User's Guide*
- *File eWay Intelligent Adapter User's Guide*
- *HTTP(S) eWay Intelligent Adapter User's Guide*
- *Oracle eWay Intelligent Adapter User's Guide*
- *SeeBeyond ICAN Suite Installation Guide*
- *SQL Server eWay Intelligent Adapter User's Guide*

1.6 SeeBeyond Web Site

The SeeBeyond Web site is your best source for up-to-the-minute product news and technical support information. The site's URL is:

http://www.seebeyond.com

1.7 SeeBeyond Documentation Feedback

We appreciate your feedback. Please send any comments or suggestions regarding this document to:

docfeedback@seebeyond.com

Chapter 2

Installing the RFID eWay

What's in This Chapter

- **"Supported Operating Systems" on page 12**
- **"System Requirements" on page 12**
- **"External System Requirements" on page 13**
- **"Installing the eWay Product Files" on page 14**
- **"Configuring Third-Party Event Managers" on page 14**

2.1 Supported Operating Systems

The RFID eWay is available on the following operating systems:

- Windows 2000, Windows XP, and Windows Server 2003
- HP-UX 11.0, 11i (PA-RISC), and 11i v2.0 (11.23)
- IBM AIX 5.1L and 5.2
- Red Hat Enterprise Linux AS 2.1 (Intel x86)
- Sun Solaris 8 and 9

2.2 System Requirements

The system requirements for the RFID eWay are the same as for eGate Integrator. Refer to the *SeeBeyond ICAN Suite Installation Guide* for a complete listing of system requirements. Review the ICAN Suite Readme file for additional requirements prior to installation.

2.2.1 Prerequisite ICAN Modules

The RFID eWay also requires that certain ICAN products be installed prior to installing the RFID eWay product files. The following table lists the prerequisite products and explains the conditions that require these products.

Table 3 Prerequisite ICAN Products

ICAN Product	Purpose in RFID Project
HTTP eWay Intelligent Adapter	The HTTP eWay is required if your RFID Project uses HTTP input (rather than JMS).
Database eWays (one or more): ▪ Oracle eWay Intelligent Adapter ▪ DB2 eWay Intelligent Adapter ▪ SQL Server eWay Intelligent Adapter	A database eWay is required if your RFID Project sends the output data to Oracle, DB2, SQL Server, or TigerLogic tables (rather than JMS). Additionally, the EPC Event Generator and sample EPC IS application require the Oracle eWay.
eInsight Business Process Manager and eVision Studio	To use the Reporting feature, you must install eInsight and eVision.
ASC X12 OTD Library (version 4010)	To use the EDI feature, you must install the ASC X12 OTD Library (version 4010).

2.2.2 JMS Requirements

To publish messages to the RFID eWay via JMS, you must install the files shown below. These files must be in a location where they will be recognized by the CLASSPATH of your Logical Host.

Table 4 JMS Requirements

File Name	Source
com.stc.jms.stcjms.jar	**logicalhost\stcis\lib** or **repository\data\files\InstallManager\50Base\stcas\common\lib** in your ICAN system
providerutil.jar **fscontext.jar** **jms.jar** **log4j.jar**	**<enterprise_designer_root>\edesigner\userdir\modules\ext\rfidadapter\j2ee**

2.3 External System Requirements

The RFID eWay supports the following external systems:

- Sun Java System RFID 1.0
- ConnecTerra RFTagAware 1.0.2

For full information on the requirements for these products, see the products' documentation.

Regarding DB2 Universal Database

Throughout this document, the terms DB2 or DB2 UDB are used to refer to DB2 Universal Database for Windows or UNIX.

2.4 Installing the eWay Product Files

The installation process includes:

- Installing the ICAN Repository.
- Uploading products to the Repository (including the RFID eWay, documentation, sample files, and Javadocs).
- Downloading components (including the Enterprise Designer and Logical Host) from the Repository.
- Updating products in the Enterprise Designer using the Update Center Wizard.

To install the RFID eWay

1. Follow the instructions for installing ICAN in the *SeeBeyond ICAN Suite Installation Guide*.
2. After uploading **eGate.sar** to the Repository, upload the following additional product files:

Important: *Before you install the RFID eWay product files, make sure you install the product files for any of the prerequisite products you plan to use. For more details on these prerequisites, see* **"Prerequisite ICAN Modules" on page 13**.

- **RFIDeWay.sar** (to install the RFID eWay)
- **RFIDeWayDocs.sar** (to install the RFID eWay documentation)

Note: *These files may not be located on the same installation CD as the* ***eGate.sar*** *file.*

2.5 Configuring Third-Party Event Managers

In addition to installing the RFID eWay product files in your ICAN system, you must also configure your third-party event managers. The following sections explain how to configure your Sun RFID and ConnecTerra systems to work with the RFID eWay.

2.5.1 Configuring the Sun RFID Event Manager

The following steps explain how to configure your Sun Java System RFID Event Manager. These instructions assume that you have already installed Sun Java System RFID. Refer to the Sun Java System RFID documentation for installation instructions.

These instructions assume that you used the default installation options and installed the application to the **SUNWrfid** directory.

Configuring your Java RFID Suite

1 Copy the files listed in Table 4 on page 13 to the **opt/SUNWrfid/lib/utils** directory.

2 Edit the **Rfid.xml** and **RfidConfig.xml** files in the **/etc/opt/SUNWrfid** directory.

In **Rfid.xml**, add the following bolded elements to the ImplementationClass tag of Rfid ServiceBean if they do not already exist:

```
<ImplementationClass>com.sun.autoid.reader.ReaderImpl
   <Resources>
      <JAR>autoid.jar</JAR>
      ...
      <JAR>com.stc.jms.stcjms.jar</JAR>
      <JAR>log4j.jar</JAR>
      <JAR>providerutil.jar</JAR>
   </Resources>
</ImplementationClass>
```

In **RfidConfig.xml**, all of the default adapter sections have been commented out and PMLReader is used. This is used by the simulator. Notice that the default port number being used is 9011. You can modify this if that port is already in use on your system. Make sure to start the simulator with the right port number. In addition, a new logger section is specified to include JMSLogger. You must alter the JndiProviderURL in this section, which points to the location of the JMS binding file.

3 Modify **RfidConfig.xml** by adding the following JMSLogger section:

```
<ems:logger>
   <ems:name>JMSLogger</ems:name>
   <ems:classname>com.sun.autoid.logger.JMSLogger</ems:classname>
      <ems:properties>
         <ems:property>LogLevel</ems:property>
         <ems:value>ALL</ems:value>
      </ems:properties>
      <ems:properties>
         <ems:property>JndiProviderURL</ems:property>
         <ems:value>file:///home/jlong/SUNWautoid/binding/.
            </ems:value>
      </ems:properties>
      <ems:properties>
         <ems:property>ConnectionFactory</ems:property>
         <ems:value>queue</ems:value>
      </ems:properties>
      <ems:properties>
         <ems:property>QueueName</ems:property>
         <ems:value>jndiQueueName</ems:value>
      </ems:properties>
</ems:logger>
```

The bolded element in the section is the location where the JMS binding file is saved. Modify it to fit your environment.

The JMSLogger section is referenced in the `<ems:filter>` tag as follows:

```
<ems:filter>
   <ems:name>RfidDelta</ems:name>
   <ems:classname>com.sun.autoid.filter.Delta</ems:classname>
   <ems:properties>
      <ems:property>LogLevel</ems:property>
      <ems:value>CONFIG</ems:value>
   </ems:properties>
   <ems:outputs>
      <ems:output>RfidFileLogger</ems:output>
      <ems:output>EpcGuiLogger</ems:output>
      <ems:output>JMSLogger</ems:output>
   </ems:outputs>
</ems:filter>
```

4 Optionally change the location where the output file will be written by the logger. To change this, locate the DemokitFileLogger section and change the following value:

```
/tmp/DemokitFileLogger.out
```

to the correct location for your log file.

Refer to **"Using the JMS binding utility"** below for further configuration information.

2.5.2 Configuring the Sun Event Manager with RFID readers

Two RFID readers are currently compatible with the Sun RFID Event Manager. They are:

- Alien ALR 9780 with firmware version 2.2.21
- Matrics AR 400 with firmware version 3.7.3

You will need to modify the **RfidConfig.xml** file to include the Matrics or Alien reader configuration (under **/etc/opt/SUNWrfid** if you install it using the default option). PMLReader is the default reader. In the source code, comment it out and uncomment the section corresponding to either the Matrics or Alien reader.

For the Matrics AR 400 reader:

1 Select the correct hostname and port number.

2 Since Matrics does not use scanDuration, add the timeout section setting.

3 Use **MatricsAR400.jar** and modify the class name to **com.sun.autoid.adapter.matrics.AR400.MatricsReaderAdapter**.

Make sure to copy **MatricsAR400.jar** to the directory **/opt/SUNWrfid/lib/utils**.

```
<ems:adapter>
    <ems:name>Matrics Reader</ems:name>
    <ems:classname>
com.sun.autoid.adapter.matrics.AR400.MatricsReaderAdapter </
ems:classname>
    <ems:properties>
        <ems:property>LogLevel</ems:property>
        <ems:value>ALL</ems:value>
    </ems:properties>
    <ems:properties>
        <ems:property>ConnectionType</ems:property>
        <ems:value>network</ems:value>
    </ems:properties>
    <ems:properties>
        <ems:property>hostname</ems:property>
        <ems:value>10.18.83.60</ems:value>
    </ems:properties>
    <ems:properties>
        <ems:property>port</ems:property>
        <ems:value>3000</ems:value>
    </ems:properties>
    <ems:properties>
        <ems:property>readerepc</ems:property>
        <ems:value>urn:epc:tag:gid-96:1.1.1</ems:value>
    </ems:properties>
    <ems:properties>
        <ems:property>readerepc_ant1</ems:property>
        <ems:value>urn:epc:tag:gid-96:1.1.1</ems:value>
    </ems:properties>
    <ems:properties>
        <ems:property>readerepc_ant2</ems:property>
        <ems:value>urn:epc:tag:gid-96:1.1.2</ems:value>
    </ems:properties>
    <ems:properties>
        <ems:property>readerepc_ant3</ems:property>
        <ems:value>urn:epc:tag:gid-96:1.1.3</ems:value>
    </ems:properties>
    <ems:properties>
        <ems:property>readerepc_ant4</ems:property>
        <ems:value>urn:epc:tag:gid-96:1.1.4</ems:value>
    </ems:properties>
    <ems:properties>
        <ems:property>Node</ems:property>
        <ems:value>4</ems:value>
    </ems:properties>
    <ems:properties>
        <ems:property>Antenna1</ems:property>
        <ems:value>1</ems:value>
    </ems:properties>
    <ems:properties>
        <ems:property>Antenna2</ems:property>
        <ems:value>0</ems:value>
    </ems:properties>
    <ems:properties>
        <ems:property>Antenna3</ems:property>
        <ems:value>0</ems:value>
    </ems:properties>
    <ems:properties>
        <ems:property>Antenna4</ems:property>
        <ems:value>0</ems:value>
    </ems:properties>
    <ems:properties>
        <ems:property>Power</ems:property>
        <ems:value>C0</ems:value>
    </ems:properties>
    <ems:properties>
        <ems:property>Environment</ems:property>
        <ems:value>4</ems:value>
    </ems:properties>
    <ems:properties>
        <ems:property>CombinedAntenna</ems:property>
        <ems:value>1</ems:value>
    </ems:properties>
    <ems:properties>
        <ems:property>FilterLength</ems:property>
        <ems:value>0</ems:value>
    </ems:properties>
    <ems:properties>
        <ems:property>Filter</ems:property>
```

```
            <ems:value>0000000000000000</ems:value>
        </ems:properties>
        <ems:properties>
            <ems:property>Wait</ems:property>
            <ems:value>80</ems:value>
        </ems:properties>
        <ems:properties>
            <ems:property>debugflags</ems:property>
            <ems:value>0x4</ems:value>
        </ems:properties>
        <ems:properties>
            <ems:property>autoread</ems:property>
            <ems:value>true</ems:value>
        </ems:properties>
        <ems:properties>
            <ems:property>timeout</ems:property>
            <ems:value>500</ems:value>
        </ems:properties>
        <ems:outputs>
            <ems:output>RfidSmoother</ems:output>
            <ems:output>EpcGuiLogger</ems:output>
        </ems:outputs>
    </ems:adapter>
```

For Alien ALR 9780, select the correct hostname and port number, and specify the appropriate username and password if they have been changed.

Below is a sample configuration section:

```
<ems:adapter>
    <ems:name>AlienReader</ems:name>

<ems:classname>com.sun.autoid.adapter.alien.NanoScannerAdapter</
ems:classname>
    <ems:properties>
        <ems:property>LogLevel</ems:property>
        <ems:value>ALL</ems:value>
    </ems:properties>
    <ems:properties>
        <ems:property>hostname</ems:property>
        <ems:value>10.18.83.60</ems:value>
    </ems:properties>
    <ems:properties>
        <ems:property>readerepc</ems:property>
        <ems:value>urn:epc:tag:gid-96:1.1.1</ems:value>
    </ems:properties>
    <ems:properties>
        <ems:property>readerepcANT0</ems:property>
        <ems:value>urn:epc:tag:gid-96:1.1.1</ems:value>
    </ems:properties>
    <ems:properties>
        <ems:property>readerepcANT1</ems:property>
        <ems:value>urn:epc:tag:gid-96:1.1.2</ems:value>
    </ems:properties>
    <ems:properties>
        <ems:property>antennaSequence</ems:property>
        <ems:value>0</ems:value>
    </ems:properties>
    <ems:properties>
        <ems:property>username</ems:property>
        <ems:value>alien</ems:value>
    </ems:properties>
    <ems:properties>
        <ems:property>password</ems:property>
        <ems:value>password</ems:value>
    </ems:properties>
    <ems:properties>
        <ems:property>persisttime</ems:property>
        <ems:value>2</ems:value>
    </ems:properties>
    <ems:properties>
        <ems:property>autoread</ems:property>
        <ems:value>true</ems:value>
    </ems:properties>
    <ems:properties>
        <ems:property>scanDuration</ems:property>
        <ems:value>500</ems:value>
    </ems:properties>
     <ems:properties>
        <ems:property>cycles</ems:property>
        <ems:value>2</ems:value>
    </ems:properties>
    <ems:properties>
        <ems:property>AcquireMode</ems:property>
        <ems:value>Inventory</ems:value>
    </ems:properties>
    <ems:outputs>
        <ems:output>RfidSmoother</ems:output>
        <ems:output>EpcGuiLogger</ems:output>
    </ems:outputs>
</ems:adapter>
```

2.5.3 Running the Sun RFID Event Manager

After running JMS binding (see **“Using the JMS binding utility” on page 23**):

1 Go to the machine running the Sun RFID Event Manager and confirm that the conditions in **“Configuring your Java RFID Suite” on page 15** are true.

You will need to modify the value to point to your binding file and use the appropriate RFID reader (PMLReader, Alien ALR 9780, or Matrics AR 400). See **“Configuring the Sun Event Manager with RFID readers” on page 16**. For example:

```
<ems:property>JndiProviderURL</ems:property>
<ems:value>file:///home/rkolar//binding/.
</ems:value>
```

Starting JAVA RFID Suite

After starting the eGate project, configuring the Java RFID Suite, and creating the JMS binding file, start the Java RFID Suite as follows:

1 Set the correct value for the `display` variable.

2 Start the Java RFID Suite product by going to **/opt/SUNwrfid/bin** and run the command `startall`.

To stop the Java RFID Suite product, run the command `stopall`.

To start the simulator, go to **/opt/SUNwrfid/bin** and run the command `./PMLReader`. This command starts the simulator which sends tag messages to the Event Manager. If the product is running correctly, the table **Observation_log** in the database will begin to be populated with data when RFID is detected by the reader.

If you encounter any problems, check the log files under **/var/opt/SUNWrfid/logs** or **stc_is_IntegrationSvr1.log**.

For further information on the Sun RFID Event Manager, see the *Sun Java System RFID Software Administration Guide*.

2.5.4 Configuring the ConnecTerra Event Manager

The following steps explain how to configure the ConnecTerra Event Manager. These instructions assume that you have already installed RFTagAware. Refer to the RFTagAware documentation for installation instructions.

To configure the ConnecTerra event manager

1. On the system where you installed RFTagAware, navigate to the **<connecterra_install>\etc** folder. Use a text editor to edit the **edge.props** file.
2. For JMS, add the following JMS drivers (including full class name):
   - com.connecterra.ale.notificationDriver.jms.class=com.connecterra.ale.notifytypes.JMSNotification
   - com.connecterra.ale.notificatonDriver.jms.default.namingPropertiesFile=file:///<connecterra_install>/etc/naming.props
3. If you are using JMS, you must also create or modify the **jms.options** file in the **<connecterra_install>\etc** directory to include one of the following statements (depending on whether you use **jmsis.jar** or **com.stc.stcjms.jms.jar**):
   - `set= JMS_LIB= <JMS_jars_folder>\com.stc.jmsis.jar;fscontext.jar;jms.jar;jta.jar;log4j.jar;providerurl.jar`
   - `set= JMS_LIB= <JMS_jars_folder>\com.stc.stcjms.jms.jar;fscontext.jar;jms.jar;jta.jar;log4j.jar;providerurl.jar`

   Refer to **"Using the JMS binding utility"** below for further configuration information.

Note: *All .jar files are prefixed with the drive location.*

4. View the contents of **monitor.props** to see which port it is configured to use. Verify that this port is not in use.

   For example, to determine whether port 12345 is in use, type

   ```
   netstat -a | find "12345"
   ```
5. Use a text editor to open **naming.props**. Enter the values for **java.naming.provider.url** and **java.naming.factory.initial**.

   For example:
   - `java.naming.provider.url=file:///c:\RFID\Connecterra`
   - `java.naming.factory.initial=com.sun.jndi.fscontext.RefFSContextFactory`

Before running the event manager

1. Navigate to the **<connecterra_install>\samples\SubscribeSample** directory.
2. Run the following batch files:
   - **ClearEdgeServerState.bat**
   - **EdgeServer.bat**
   - **MonitorConsole.bat** (optional)

2.5.5 Configuring the ConnecTerra Event Manager with RFID Readers

The ConnecTerra RFTagAware Edge Server 1.1.2 is currently compatible with the following RFID readers:

- Alien ALR 9780 with firmware version 2.2.21
- Matrics AR 400 with firmware version 3.7.3

To connect a reader to ConnecTerra RFTagAware, the **edge.props** file needs to contain the following sections:

```
# For Alien ALR9780 Reader
com.connecterra.ale.reader.AlienALR9780Readr1.class =
com.connecterra.ale.readertypes.AlienALR9780PhysicalReader
com.connecterra.ale.reader.AlienALR9780Readr1.hostname = <Reader IP
address>
com.connecterra.ale.reader.AlienALR9780Readr1.username = <Reader user
name>
com.connecterra.ale.reader.AlienALR9780Readr1.password = <Reader
password>
com.connecterra.ale.reader.AlienALR9780Readr1.defaultRate = <The
period (in ms) between the start of one read cycle and the start of
the next>
com.connecterra.ale.reader.AlienALR9780Readr1.uhf1LogicalReaderName =
<Logical reader name>

# For Matrics AR400 Reader
com.connecterra.ale.reader.MatricsAR400Readr1.class =
com.connecterra.ale.readertypes.MatricsAR400PhysicalReader
com.connecterra.ale.reader.MatricsAR400Readr1.hostname = <Reader IP
address>
com.connecterra.ale.reader.MatricsAR400Readr1.defaultRate = <The
period (in ms) between the start of one read cycle and the start of
the next>
com.connecterra.ale.reader.MatricsAR400Readr1.uhf1LogicalReaderName =
<Logical reader name>
```

As an example, the Alien ALR 9780 can be modified as follows:

```
# For Alien ALR9780 Reader
com.connecterra.ale.reader.AlienALR9780Readr1.class =
com.connecterra.ale.readertypes.AlienALR9780PhysicalReader
com.connecterra.ale.reader.AlienALR9780Readr1.hostname =
192.168.127.234
com.connecterra.ale.reader.AlienALR9780Readr1.username = alien
com.connecterra.ale.reader.AlienALR9780Readr1.password = password
com.connecterra.ale.reader.AlienALR9780Readr1.defaultRate = 1
com.connecterra.ale.reader.AlienALR9780Readr1.uhf1LogicalReaderName =
AlienLogicalReader1
```

The Matrics AR 400 can be modified as follows:

```
# For Matrics AR400 Reader
com.connecterra.ale.reader.MatricsAR400Readr1.class =
com.connecterra.ale.readertypes.MatricsAR400PhysicalReader
com.connecterra.ale.reader.MatricsAR400Readr1.hostname =
192.168.127.254
com.connecterra.ale.reader.MatricsAR400Readr1.defaultRate = 1
com.connecterra.ale.reader.MatricsAR400Readr1.uhf1LogicalReaderName =
MatricsLogicalReader1
```

When running the application, make sure the Event Cycle specification file contains the correct logical reader name. For example, the **ECSpec.xml** file should contain the following:

```
<EventCycleSpec xmlns="http://schemas.connecterra.com/ale">

    <logicalReaders>
        <!-- determines which logical reader(s) will contribute to
this event cycle.  Logical reader names are defined in edge.props. -->
        <logicalReader>MatricsLogicalReader1</logicalReader>
    </logicalReaders>
```

For the Matrics AR 400 reader, add the correct properties for the “Read Point Class,” as follows:

- Type: `area`
- Scan Period: `on demand`
- Gained: `100%`
- Retry: `3`
- Air protocol: `all`
- Tag Selection: `all`
- Write Gain: `100%`

2.5.6 Running the ConnecTerra Event Manager using HTTP

1. From a command prompt, navigate to the **<connecterra_install>\samples\SubscribeSample** directory.
2. Make sure the `<logicalReader>` tag in ECSpec.xml has the correct value to match the logical reader name in **edge.props**.
3. Type the following command:

   ```
   run define <ECSpec name> ECSpec.xml
   ```

   For example:

   ```
   run define EC1 ECSpec.xml
   ```

4. Type the following command:

   ```
   run subscribe <ECSpec name> <my_servlet_URL>
   ```

   For example:

   ```
   run subscribe EC1 http://localhost:18004/Deployment1_servlet/
   SBYNConnecTerraServlet
   ```

To stop your HTTP event manager

1. From a command prompt, navigate to the **<connecterra_install>\samples\SubscribeSample** directory.
2. Type the following command:

   ```
   run unsubscribe <ECSpec name> <my_servlet_URL>
   ```

   For example:

   ```
   run unsubscribe EC1 http://localhost:18004/Deployment1_servlet/
   SBYNConnecTerraServlet
   ```
3. Type the following command:

   ```
   run stop <ECSpec name>
   ```

   For example:

   ```
   run stop EC1
   ```

2.5.7 Running Your ConnecTerra Event Manager Using JMS

1. From a command prompt, navigate to the ConnecTerra JMS bindings folder.
2. Execute the **jmsbind** command.
3. Navigate to the subscribe sample folder and type the following commands:

   ```
   run define <myECSpec> ECSpec.xml
   run subscribe <myECSpec> jms:/queue/queue/jndiQueueName
   ```

   For example:

   ```
   run define EC1 ECSpec.xml
   run subscribe EC1 jms:/queue/queue/jndiQueueName
   ```

To stop your JMS event manager

1. Navigate to the subscribe sample folder.
2. Type the following command:

   ```
   run stop <myECSpec>
   ```

   For example:

   ```
   run stop EC1
   ```

2.5.8 Using the JMS binding utility

The JMS binding utility comprises the following components:

- The Java library **jmsbind.jar**
- The command script **jmsbind.bat**
- The configuration file **binding.properties**
- The log4j runtime configuration file **log4j.xml**

jmsbind.jar consists of the utility program **CreateBindings.** It reads in both the binding and log4j configuration files **binding.properties** and **log4j.xml** correspondingly. **CreateBindings** expects the configurations files to be in the same directory as itself.

Binding.properties consists of a list of name/value pairs, as follows:

1 JMS Server Name

   `jms.server.name.<Server ID>=` where `<Server ID>` begins with 1 in increments of 1 (for example, `jms.server.name.1=server1.stc.com`). More than one server can be specified (for example, `jms.server.name.2=server2.stc.com`). The `<Server ID>` is used to link other properties to the same server.

2 JMS Port Number

   `jms.server.port.<Server ID>=` where `<Server ID>` begins with 1 in increments of 1 (for example, `jms.server.port.1=18008`). Each server can only have one port number specified.

3 JMS Queue Connection Factory

   `jms.queue.conn.factory.<Server ID>.<Connection Factory ID>=` where `<Server ID>` begins with 1 in increments of 1, and `<Connection Factory ID>` also begins with 1 in increments of 1 (for example, `jms.queue.conn.factory.1.1=queue1`). For queue connection factory, more than one instance is allowed (for example, `jms.queue.conn.factory.1.2=queue2`).

4 JMS Topic Connection Factory

   `jms.topic.conn.factory.<Server ID>.<Connection Factory ID>=` where `<Server ID>` begins with 1 in increments of 1, and `<Connection Factory ID>` also begins with 1 in increments of 1 (for example, `jms.topic.conn.factory.1.1=topic1`). For topic connection factory, more than one instance is allowed (for example, `jms.topic.conn.factory.1.2=topic2`).

5 JMS XA Queue Connection Factory

   `jms.xa.queue.conn.factory.<Server ID>.<Connection Factory ID>=` where `<Server ID>` begins with 1 in increments of 1, and `<Connection Factory ID>` also begins with 1 in increments of 1 (for example, `jms.xa.queue.conn.factory.1.1=xaqueue1`). For XA queue connection factory, more than one instance is allowed (for example, `jms.xa.queue.conn.factory.1.2=xaqueue2`).

6 # JMS XA Topic Connection Factory

   `jms.xa.topic.conn.factory.<Server ID>.<Connection Factory ID>=` where `<Server ID>` begins with 1 in increments of 1, and `<Connection Factory ID>` also begins with 1 in increments of 1 (for example, `jms.xa.topic.conn.factory.1.1=xatopic1`). For XA topic connection factory, more than one instance is allowed (for example, `jms.xa.topic.conn.factory.1.2=topic2`).

7 JMS Queue Name

   This is the queue name used in the ICAN Suite **Connectivity Map**. `jms.queue.name.<Destination ID>=` where `<Destination ID>` begins with 1 in increments of 1 (for example, `jms.queue.name.1=myqueue1`). More than one instance of JMS queue name is allowed (for example, `jms.queue.name.2=myqueue2`).

8 JMS JNDI Queue Name

This is the JNDI name for the queue specified by the JMS queue name. `jms.queue.jndiname.<Destination ID>=` the `<Destination ID>` of this JNDI name must match the same value in the queue name (for example, if the JMS queue name has the name/value pair `jms.queue.name.1=myqueue1`, the JNDI queue name should be `jms.queue.jndiname.1=myjndiname1`).

9 JMS Topic Name

This is the topic name used in the ICAN Suite **Connectivity Map**. `jms.topic.name.<Destination ID>=` where `<Destination ID>` begins with 1 in increments of 1 (for example, `jms.topic.name.1=mytopic1`). More than one instance of JMS topic name is allowed (for example, `jms.topic.name.2=mytopic2`).

10 JMS JNDI Topic Name

This is the JNDI name for the topic specified by the JMS topic name. `jms.topic.jndiname.<Destination ID>=` the `<Destination ID>` of this JNDI name must match the same value in the topic name (for example, if the JMS name name has the name/value pair `jms.topic.name.1=mytopic1`, the JNDI topic name should be `jms.topic.jndiname.1=myjnditopic1`).

Once **binding.properties** is setup properly, one needs to verify that the classes required in the **jmsbind.bat** file are in the classpath, and that they are copied to the correct directories.

To run the utility, type **jmsbind**. The components that are bound will be displayed.

Chapter 3

Configuring the RFID eWay

What's in This Chapter

- "Creating RFID eWay Projects" on page 26
- "Sun RFID Projects" on page 28
- "ConnecTerra Projects" on page 49

3.1 Creating RFID eWay Projects

The RFID eWay uses a wizard to create a Project template for you to use with your RFID system. After using the Wizard to create the Project, you can then configure it for your environment, create a Deployment Profile, and activate the Project. This process is relatively simple and requires only a minimal amount of configuration on your part.

To create a new RFID Application

1. Right-click your Repository and then click **New Project**. This creates the new Project that you will use to contain your RFID application.
2. Rename the container Project if you wish.
3. Right-click the new Project, point to **New**, and then click **New RFID Application**.

   The New RFID Application Wizard appears.
4. Select the appropriate **Vendor**, **Input Type**, and **Output Type** for your system. Optionally, select **Report** and/or **EDI** if you plan to use these features.
5. Click **Finish** when you are done.
6. The new RFID application appears—as one or more Projects within the container Project—after the wizard is finished.

The Project name will indicate the type of RFID application you selected, and may also include additional Projects. See the following table for a list of Project template types created by the wizard.

Table 5 Selecting a Project Template

RFID Vendor	Input	Output	See This Section
Sun RFID	JMS	Oracle	**Using the SunRFIDSuiteJMSOracleTemplate Project** on page 29
		DB2	**Using the SunRFIDSuiteJMSDB2Template Project** on page 37
		SQL Server	**Using the SunRFIDSuiteJMSSQLServerTemplate Project** on page 40
		TigerLogic	**Using the SunRFIDSuiteJMSTigerLogicTemplate Project** on page 43
ConnecTerra	JMS	Oracle	**Using the ConnecterraJMSOracleTemplate Project** on page 50
		DB2	**Using the ConnecterraJMSDB2Template Project** on page 53
		SQL Server	**Using the ConnecterraJMSSQLServerTemplate Project** on page 56
		TigerLogic	**Using the ConnecterraJMSTigerLogicTemplate Project** on page 59
	HTTP	JMS	**Using the HTTPJMSQueueTemplate Project** on page 65
		Oracle	**Using the ConnecterraHTTPOracleTemplate Project** on page 69
		DB2	**Using the ConnecterraHTTPDB2Template Project** on page 72
		SQL Server	**Using the ConnecterraHTTPSQLServerTemplate Project** on page 75
		TigerLogic	**Using the ConnecterraHTTPTigerLogicTemplate Project** on page 78

3.1.1 Using the Reports and EDI Options

The RFID Application Wizard includes two options for enabling Reporting and EDI. As the wizard shows, eInsight and eVision are prerequisites for the Report option, and X12 is the prerequisite for the EDI option.

Selecting either or both of these options causes the wizard to create additional Projects as it is creating the RFID application in your container Project. For more information on Reports, see **Chapter 6**. For more information on EDI, see **Chapter 8**.

3.2 Sun RFID Projects

The following sections provide information for using the Sun RFID Event Manager Project templates. These topics include:

- **"Using the SunRFIDSuiteJMSOracleTemplate Project" on page 29**
- **"Using the SunRFIDSuiteJMSDB2Template Project" on page 37**
- **"Using the SunRFIDSuiteJMSSQLServerTemplate Project" on page 40**
- **"Using the SunRFIDSuiteJMSTigerLogicTemplate Project" on page 43**

3.2.1 Using the SunRFIDSuiteJMSOracleTemplate Project

This Project uses the Sun RFID Event Manager, JMS input, and Oracle Output. To use this Project template you will complete the following general steps:

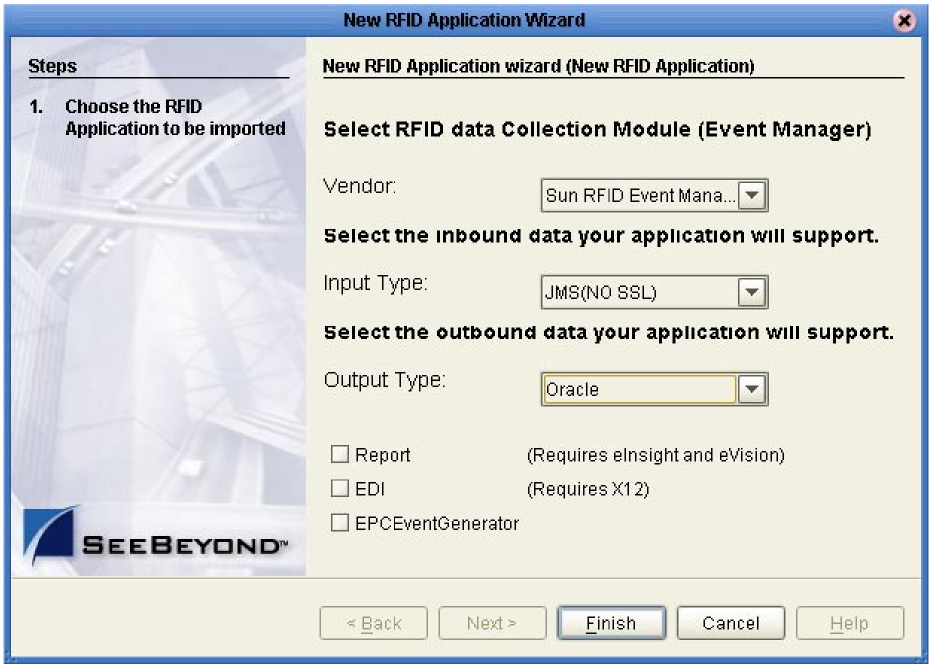
- Run the New RFID Application Wizard
- Create an Environment.
- Configure the Oracle connection.
- Configure the JMS Topic.
- Create and activate a Deployment Profile.

Create the SunRFIDSuiteJMSOracleTemplate Project template

1. Create a new Project in your Repository. This Project will act as the container Project for this RFID application.
2. Right-click the container Project, point to **New**, and then click **New RFID Application**.

   The New RFID Application Wizard appears.
3. Select the options shown in the following figure, and then click **Finish**.

Figure 1 SunRFIDSuiteJMSOracleTemplate Options



New RFID Application Wizard

Steps

1. Choose the RFID Application to be imported

New RFID Application wizard (New RFID Application)

Select RFID data Collection Module (Event Manager)

Vendor: Sun RFID Event Mana...

Select the inbound data your application will support.

Input Type: JMS(NO SSL)

Select the outbound data your application will support.

Output Type: Oracle

□ Report (Requires eInsight and eVision)

□ EDI (Requires X12)

□ EPCEventGenerator

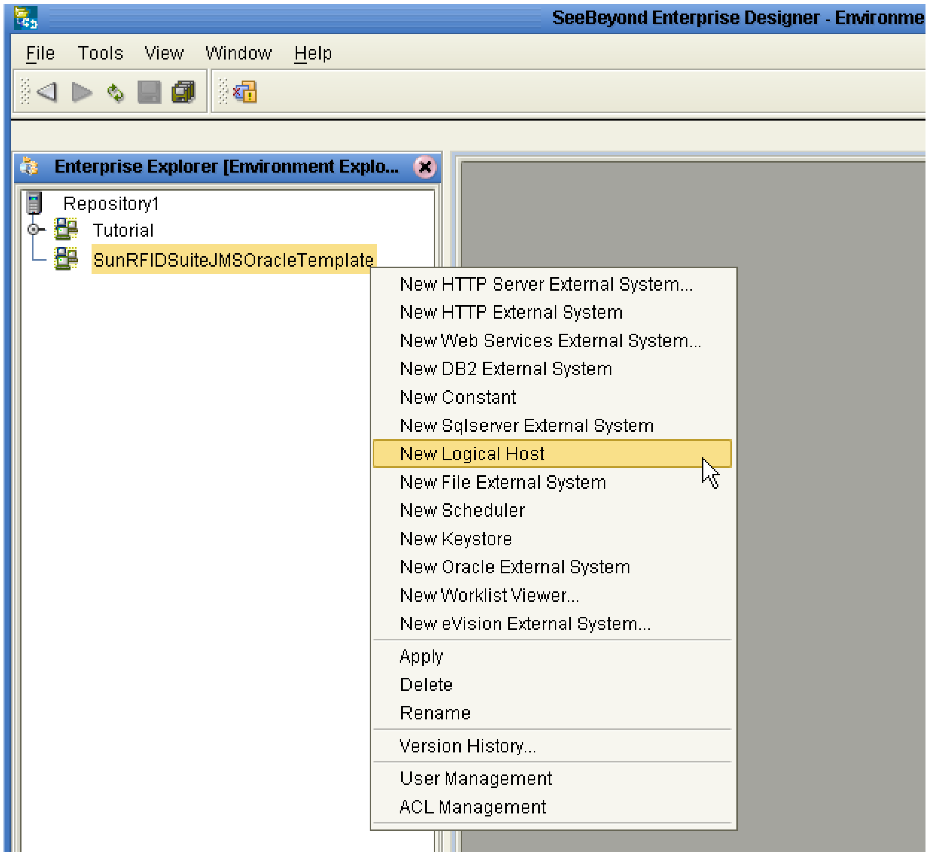
The New RFID Application Wizard creates a new Project template called **SunRFIDSuiteJMSOracleTemplate** in your container Project.

Create and configure the Environment

1. In the Enterprise Explorer pane, click the Environment Explorer tab.
2. Right-click your Repository, and then click **New Environment**.
3. Rename this Environment so it can be easily identified when you create the Deployment Profile (for example, **SunRFIDSuiteJMSOracleTemplate**).
4. To add a new Logical Host, right-click your new Environment **SunRFIDSuiteJMSOracleTemplate** and then click **New Logical Host**.

   A new Logical Host with the default name **LogicalHost1** appears under your new environment.

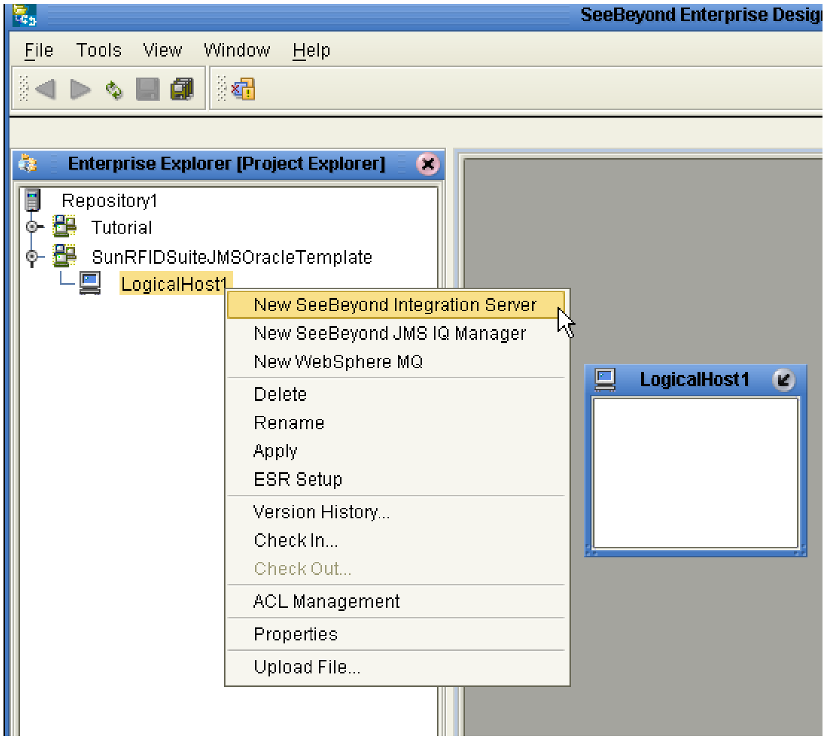
Figure 2 New Logical Host


SeeBeyond Enterprise Designer - Environme
File
Tools
View
Window
Help
Enterprise Explorer [Environment Explo...
Repository1
Tutorial
SunRFIDSuiteJMSOracleTemplate
New HTTP Server External System...
New HTTP External System
New Web Services External System...
New DB2 External System
New Constant
New Sqlserver External System
New Logical Host
New File External System
New Scheduler
New Keystore
New Oracle External System
New Worklist Viewer...
New eVision External System...
Apply
Delete
Rename
Version History...
User Management
ACL Management

5 To add a new SeeBeyond Integration Server, right-click your new Logical Host **Logical Host1** (default name) and click **New SeeBeyond Integration Server**.

A New SeeBeyond Integration Server with the default name **Integration Svr1** appears under your Logical Host.

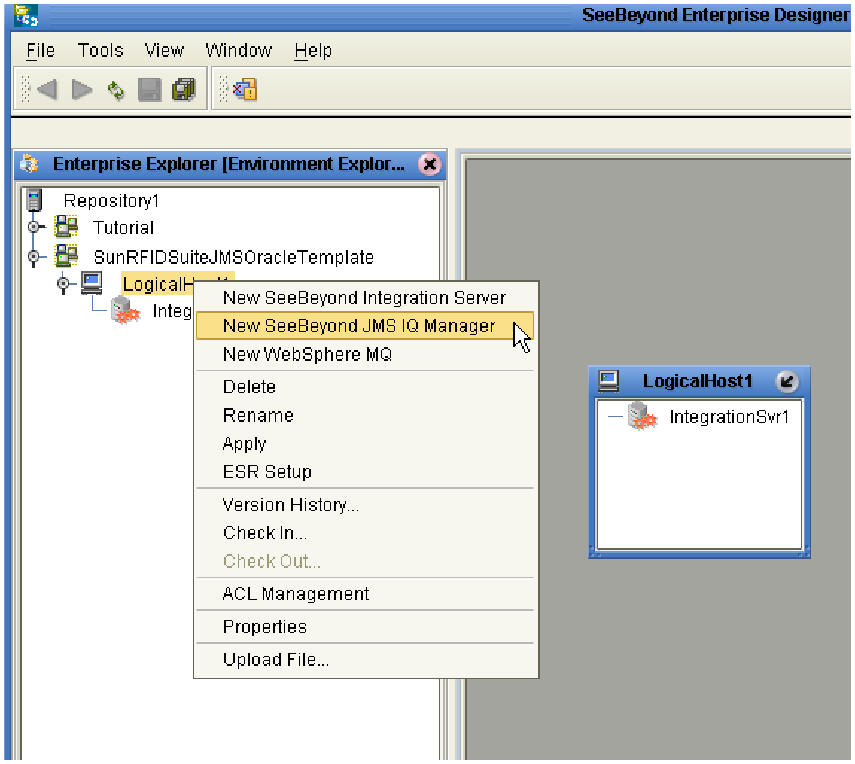
Figure 3 New SeeBeyond Integration Server



6 To add a new SeeBeyond JMS IQ Manager, right-click your Logical Host **Logical Host1** (default name) and click **New SeeBeyond JMS IQ Manager**.

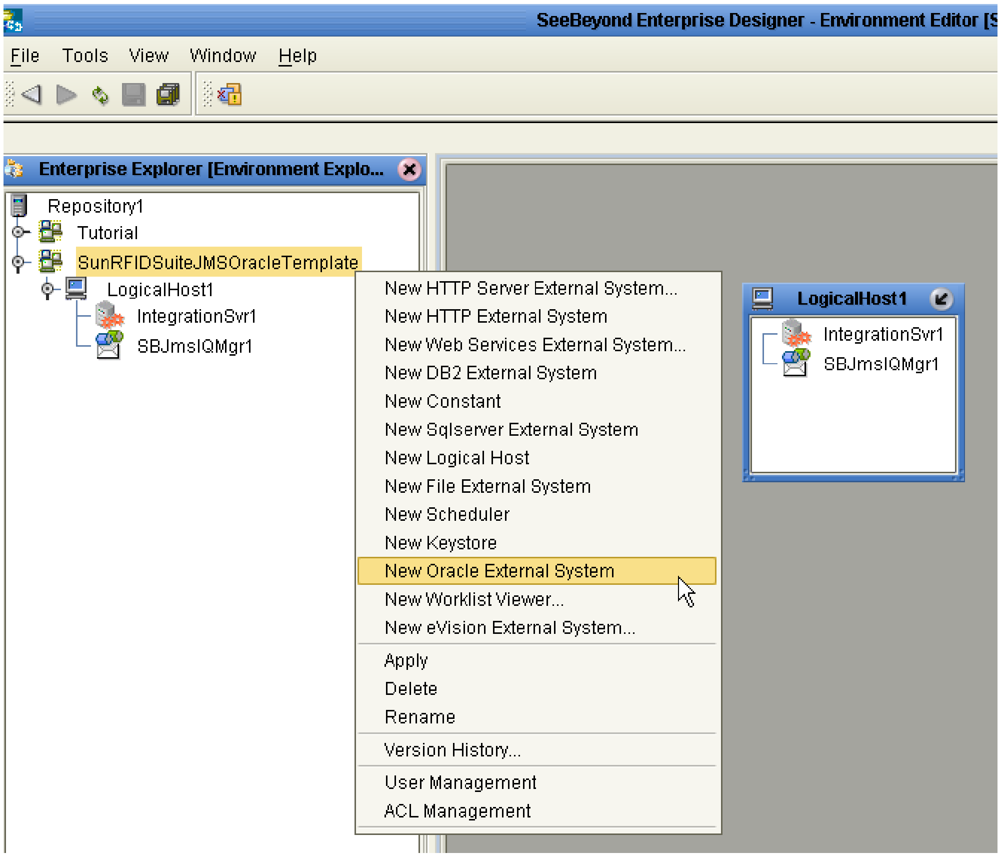
A New SeeBeyond JMS IQ Manager with the default name **SBJmsIQMgr1** appears under your Logical Host.

Figure 4 New SeeBeyond JMS IQ Manager



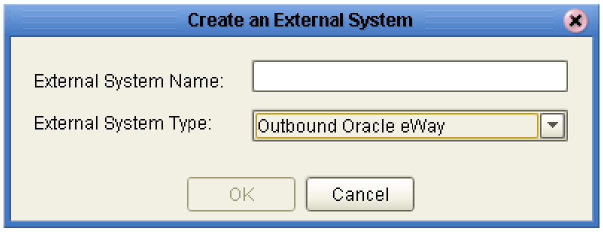
7 To add a new Oracle External System, right-click your Environment **SunRFIDSuiteJMSOracleTemplate**, point to **New**, and then click **New Oracle External System**.

Figure 5 New Oracle External System



The following menu appears.

Figure 6 New Oracle External System Menu



Create an External System

External System Name:

External System Type: Outbound Oracle eWay

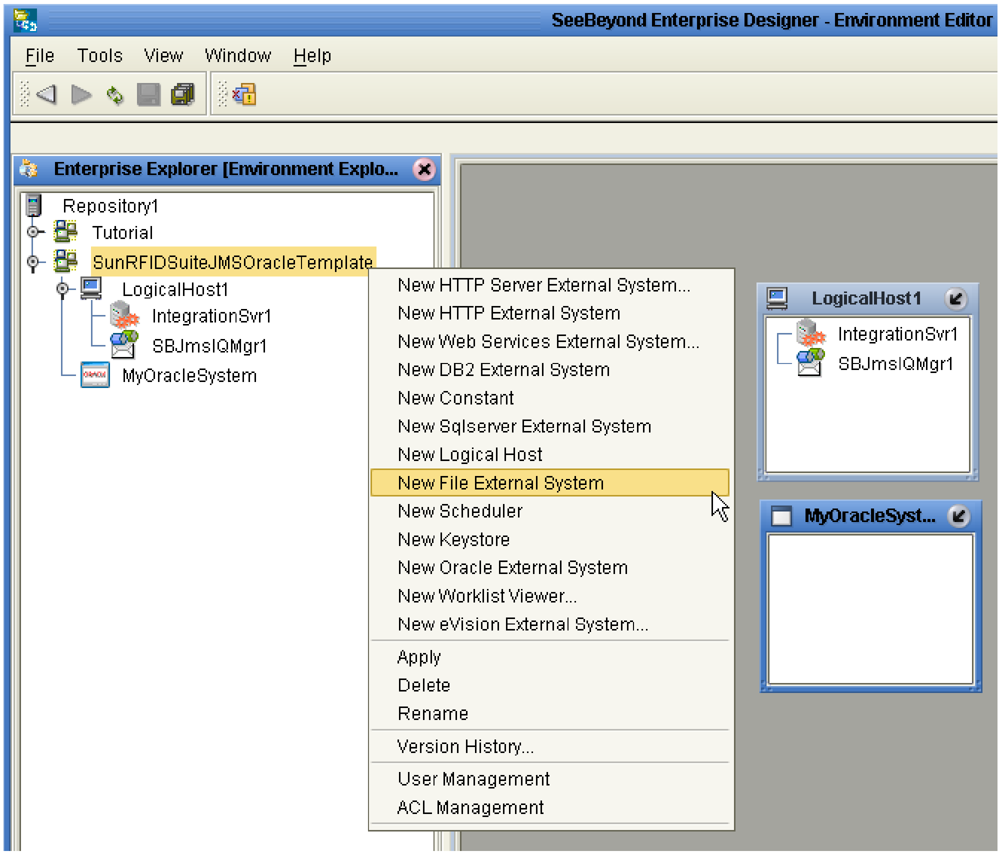
8 In the **External System Name** field, type **MyOracleSystem.**

9 From the **External System Type** drop-down menu, select **Outbound Oracle eWay**, and then click **OK.**

A new Oracle External System with the name **MyOracleSystem** appears under your Environment.

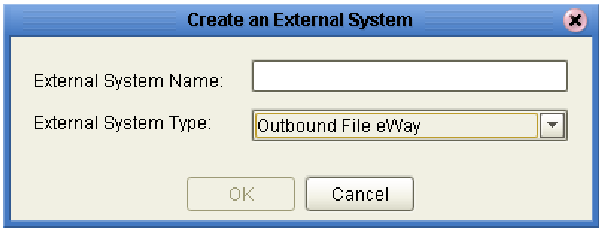
10 To add a new File External System, right-click your Environment **SunRFIDSuiteJMSOracleTemplate** and then click **New File External System**.

Figure 7 New File External System



The following menu appears.

Figure 8 New File External System Menu


Create an External System

External System Name:

External System Type: Outbound File eWay

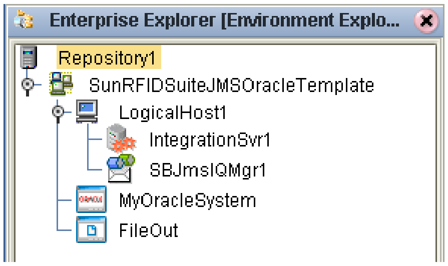
11 In the **External System Name** field, type **FileOut.**

12 From the **External System Type** drop-down menu, select **Outbound File eWay**, and then click **OK.**

A new File External System with the name **FileOut** appears under your Environment.

Your Environment looks like the following figure.

Figure 9 SunRFIDSuiteJMSOracleTemplate Environment


Enterprise Explorer [Environment Explo...
Repository1
SunRFIDSuiteJMSOracleTemplate
LogicalHost1
IntegrationSvr1
SBJmsIQMgr1
MyOracleSystem
FileOut

13 Right-click **MyOracleSystem**, and then click **Properties**.

14 Configure the Oracle External System as appropriate, and click **OK** when finished.

For complete details on configuring the outbound Oracle System, see the *Oracle eWay Intelligent Adapter User's Guide*.

Configure the JMS Topic

The JMS Topic requires no configuration. However, the name of the JMS Topic in the Connectivity Map must match the name of the topic to which your Sun RFID Event Manager is publishing. To configure the inbound Topic, you must either rename it in the Connectivity Map or in your Sun RFID system. They both must match.

Note: *To rename the Topic in the Connectivity Map, you must first check it out from the Repository. Make sure to check it back in when you are done changing the name.*

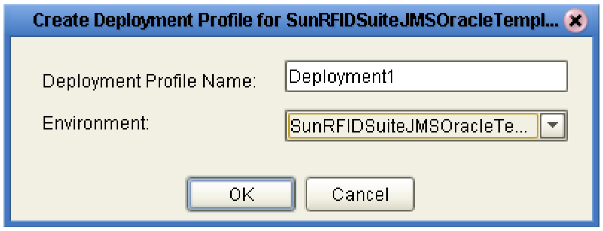
Create the Deployment Profile

1 In the Enterprise Explorer pane, click the Project Explorer tab.

2 Right-click the **SunRFIDSuiteJMSOracleTemplate** Project, point to **New**, and then click **Deployment Profile**.

The following menu appears.

Figure 10 Deployment Profile Menu



Deployment Profile Name: Deployment1

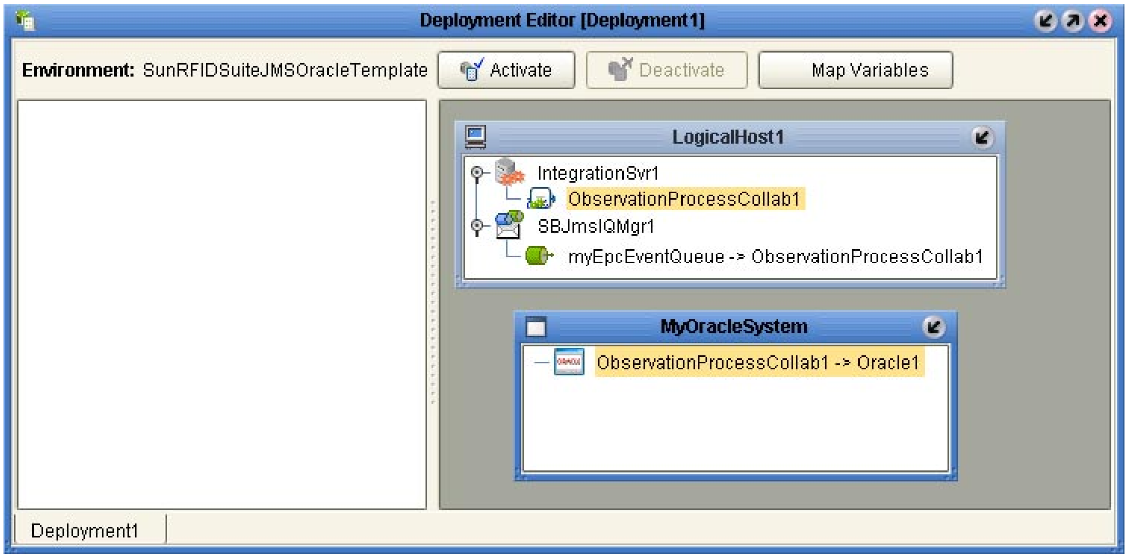
Environment: SunRFIDSuiteJMSOracleTe...

3 Type a **Deployment Profile Name**, select the appropriate **Environment**, and then click **OK**.

Select the Environment that you created in steps 2 and 3 of the *Create and configure the Environment* procedure.

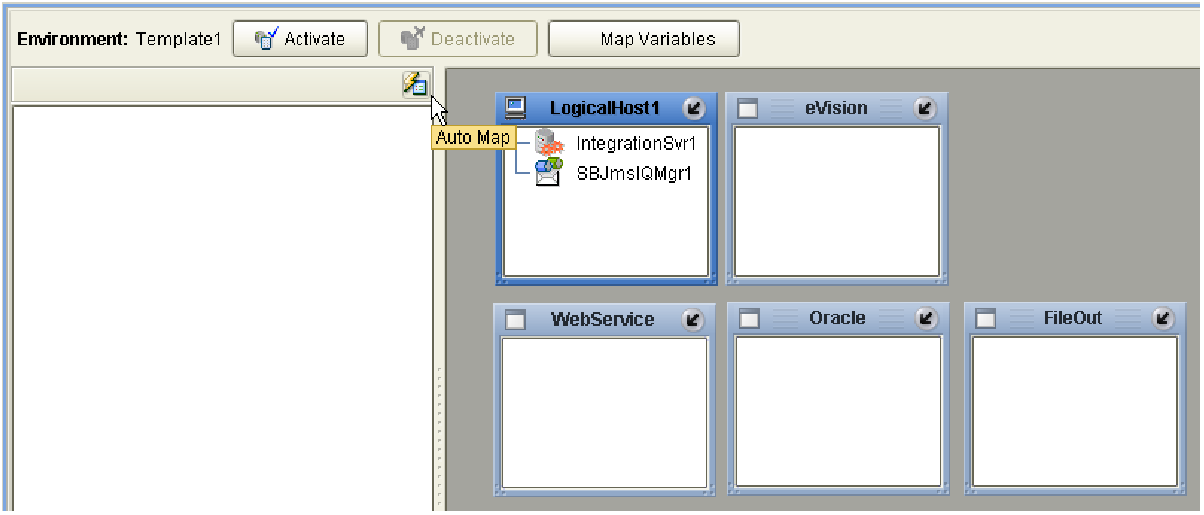
4 In the Deployment Profile, drag the components to the appropriate location, as shown in the following figure.

Figure 11 Completed Deployment Profile


Deployment Editor [Deployment1]
Environment: SunRFIDSuiteJMSOracleTemplate
Activate
Deactivate
Map Variables
LogicalHost1
IntegrationSvr1
ObservationProcessCollab1
SBJmsIQMgr1
myEpcEventQueue -> ObservationProcessCollab1
MyOracleSystem
ObservationProcessCollab1 -> Oracle1
Deployment1

You may also use the **Auto Map** feature in **Enteprise Designer**'s **Environment Template**, by clicking the **AutoMap** button, to complete your Deployment Profile.

Figure 12 Auto Map Option



5 Click **Activate** to start the activation process.

6 Since your Logical Host is not yet running, click **No** to complete the activation without applying the changes to the Logical Host.

3.2.2 Using the SunRFIDSuiteJMSDB2Template Project

This Project uses the Sun RFID Event Manager, JMS input, and DB2 UDB Output. To use this Project template, complete the following general steps:

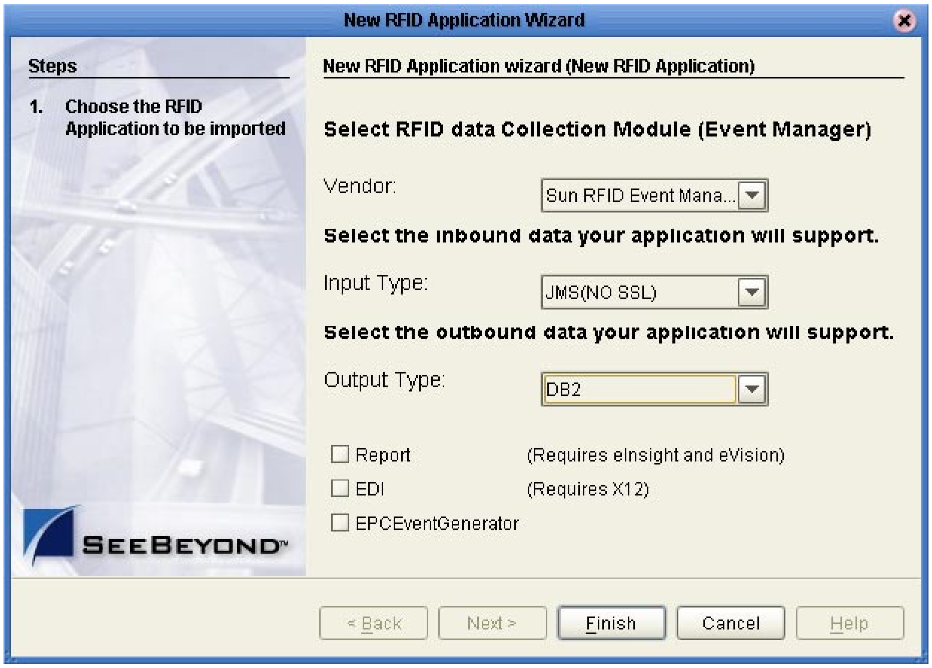
- Run the New RFID Application Wizard
- Create an Environment.
- Configure the DB2 connection.
- Configure the JMS Topic.
- Create and activate a Deployment Profile.

Create the SunRFIDSuiteJMSDB2Template Project template

1. Create a new Project in your Repository. This Project will act as the container Project for this RFID application.
2. Right-click the contain Project, point to **New**, and then click **New RFID Application**.

   The New RFID Application Wizard appears.
3. Select the options shown in the following figure, and then click **Finish**.

Figure 13 SunRFIDSuiteJMSDB2Template Options



New RFID Application Wizard

Steps

1. Choose the RFID Application to be imported

New RFID Application wizard (New RFID Application)

Select RFID data Collection Module (Event Manager)

Vendor: Sun RFID Event Mana...

Select the inbound data your application will support.

Input Type: JMS(NO SSL)

Select the outbound data your application will support.

Output Type: DB2

☐ Report (Requires eInsight and eVision)

☐ EDI (Requires X12)

☐ EPCEventGenerator

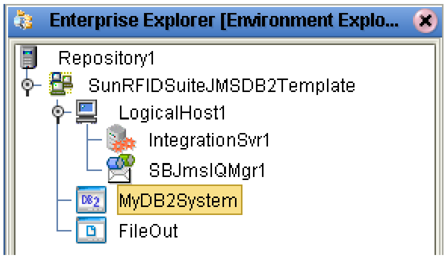
The New RFID Application Wizard creates a new Project template called **SunRFIDSuiteJMSDB2Template** in your container Project.

Create and configure the Environment

1. In the Enterprise Explorer pane, click the Environment Explorer tab.
2. Right-click your Repository, and then click **New Environment**.
3. Rename this Environment so it can be easily identified when you create the Deployment Profile, (or example **SunRFIDSuiteJMSDB2Template**).
4. Add the following items to your Environment:
   - Logical Host
   - SeeBeyond Integration Server
   - SeeBeyond JMS IQ Manager
   - DB2 External System (outbound)
   - File External System (outbound)

   Your Environment looks like the following figure.

Figure 14 SunRFIDSuiteJMSDB2Template Environment


Enterprise Explorer [Environment Explo...
Repository1
SunRFIDSuiteJMSDB2Template
LogicalHost1
IntegrationSvr1
SBJmsIQMgr1
MyDB2System
FileOut

5. Right-click **MyDB2System**, and then click **Properties**.
6. Configure the DB2 external as appropriate for your DB2 system, and click **OK** when finished.

   For complete details on configuring the outbound DB2 System, see the *DB2 Universal Database eWay Intelligent Adapter User's Guide*.

Configure the JMS Topic

The JMS Topic requires no configuration. However, the name of the JMS Topic in the Connectivity Map must match the name of the topic to which your Sun RFID Event Manager is publishing. To configure the inbound Topic, you must either rename it in the Connectivity Map or in your Sun RFID system. They both must match.

Note: *To rename the Topic in the Connectivity Map, you must first check it out from the Repository. Make sure to check it back in when you are done changing the name.*

Create Deployment Profile

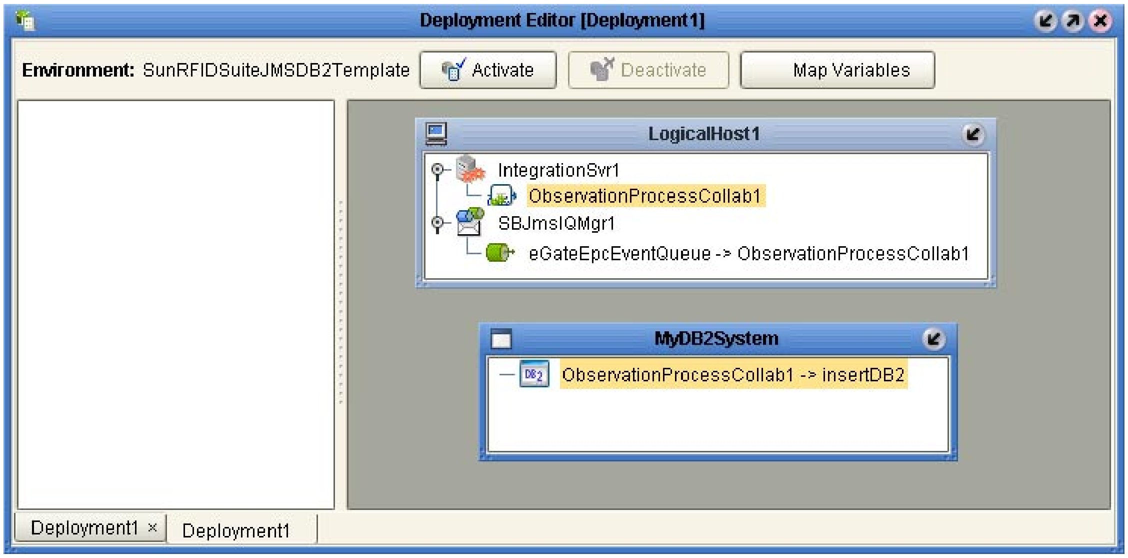
1. In the Enterprise Explorer pane, click the Project Explorer tab.
2. Right-click the **SunRFIDSuiteJMSDB2Template** Project, point to **New**, and then click **Deployment Profile**.

3 Type a **Deployment Profile Name**, select the appropriate **Environment**, and then click **OK**.

Make sure to select the Environment that you created in steps 2 and 3 of the previous procedure.

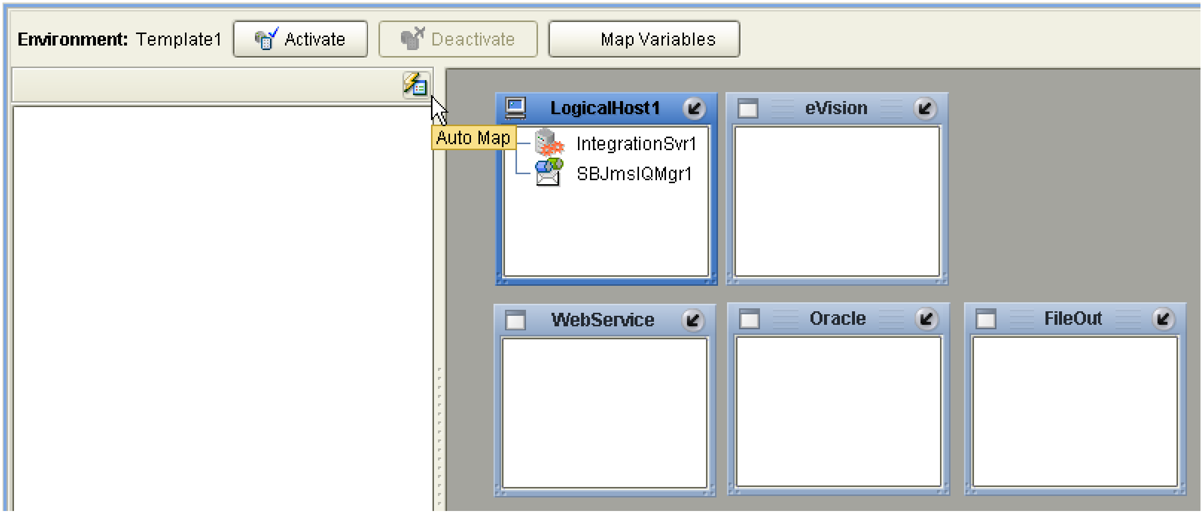
4 In the Deployment Profile, drag the components to the appropriate location as shown in the following figure.

Figure 15 Completed Deployment Profile


Deployment Editor [Deployment1]
Environment: SunRFIDSuiteJMSDB2Template
Activate
Deactivate
Map Variables
LogicalHost1
IntegrationSvr1
ObservationProcessCollab1
SBJmsIQMgr1
eGateEpcEventQueue -> ObservationProcessCollab1
MyDB2System
ObservationProcessCollab1 -> insertDB2
Deployment1 ×
Deployment1

You may also use the **Auto Map** feature in **Enteprise Designer**'s **Environment Template**, by clicking the **AutoMap** button, to complete your Deployment Profile.

Figure 16 Auto Map Option



5 Click **Activate** to start the activation process.

6 Since your Logical Host is not yet running, click **No** to complete the activation without applying the changes to the Logical Host

3.2.3 Using the SunRFIDSuiteJMSSQLServerTemplate Project

This Project uses the Sun RFID Event Manager, JMS input, and SQL Server Output. To use this Project template you will complete the following general steps:

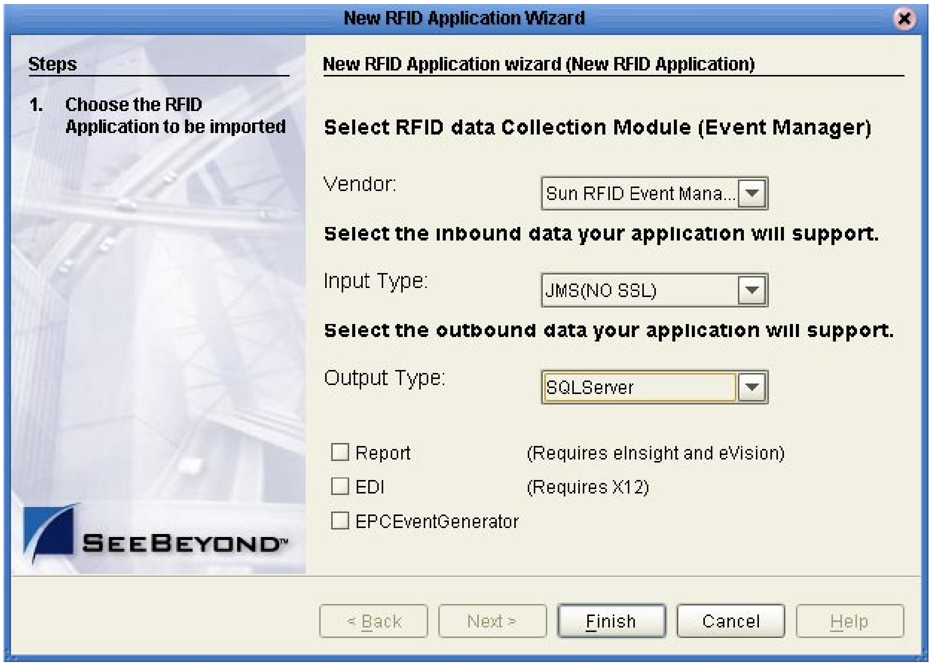
- Run the New RFID Application Wizard
- Create an Environment.
- Configure the SQL Server connection.
- Configure the JMS Topic.
- Create and activate a Deployment Profile.

Create the SunRFIDSuiteJMSSQLServerTemplate Project template

1. Create a new Project in your Repository. This Project will act as the container Project for this RFID application.
2. Right-click the container Project, point to **New**, and then click **New RFID Application**

   The New RFID Application Wizard appears.
3. Select the options shown in the following figure, and then click **Finish**.

Figure 17 SunRFIDSuiteJMSSQLServerTemplate Options



New RFID Application Wizard

Steps

1. Choose the RFID Application to be imported

New RFID Application wizard (New RFID Application)

Select RFID data Collection Module (Event Manager)

Vendor: Sun RFID Event Mana...

Select the inbound data your application will support.

Input Type: JMS(NO SSL)

Select the outbound data your application will support.

Output Type: SQLServer

☐ Report (Requires eInsight and eVision)

☐ EDI (Requires X12)

☐ EPCEventGenerator

The new RFID Application Wizard creates a new Project template called **SunRFIDSuiteJMSSQLServerTemplate** in your container Project.

Create and configure the Environment

1 In the Enterprise Explorer pane, click the Environment Explorer tab.

2 Right-click your Repository, and then click **New Environment**.

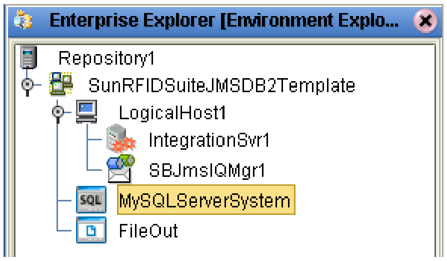
3 Rename this Environment so it can be easily identified when you create the Deployment Profile (for example, **SunRFIDSuiteJMSSQLServerTemplate**).

4 Add the following items to your Environment:

- Logical Host
- SeeBeyond Integration Server
- SeeBeyond JMS IQ Manager
- SQL Server External System (outbound)
- File External System (outbound)

Your Environment looks like the following figure.

Figure 18 SunRFIDSuiteJMSSQLServerTemplate Environment



5 Right-click **MySQLServerSystem**, and then click **Properties**.

6 Configure the SQL Server external as appropriate for your SQL Server system, and then click **OK** when finished.

For complete details on configuring the outbound SQL Server System, see the *SQL Server eWay Intelligent Adapter User's Guide*.

Configure the JMS Topic

The JMS Topic requires no configuration. However, the name of the JMS Topic in the Connectivity Map must match the name of the topic to which your Sun RFID Event Manager is publishing. To configure the inbound Topic, you must either rename it in the Connectivity Map or in your Sun RFID system. They both must match.

Note: To rename the Topic in the Connectivity Map, you must first check it out from the Repository. Make sure to check it back in when you are done changing the name.

Create Deployment Profile

1 In the Enterprise Explorer pane, click the Project Explorer tab.

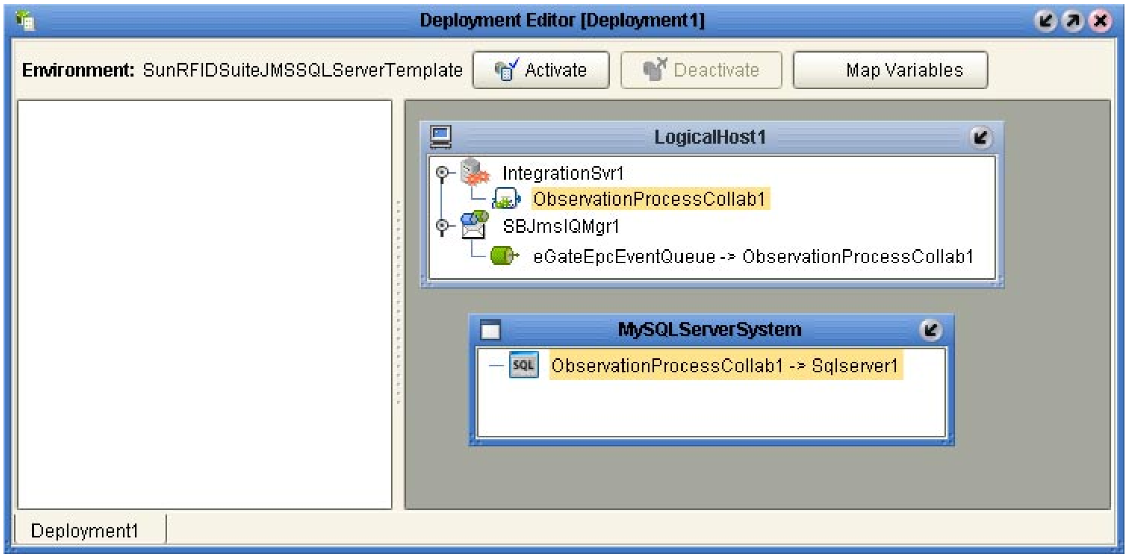
2 Right-click the **SunRFIDSuiteJMSSQLServerTemplate** Project, point to **New**, and then click **Deployment Profile**.

3 Type a **Deployment Profile Name**, select the appropriate **Environment**, and then click **OK**.

Make sure to select the Environment that you created in steps 2 and 3 of the previous procedure.

4 In the Deployment Profile, drag the components to the appropriate location as shown in the following figure.

Figure 19 SunRFIDSuiteJMSSQLServerTemplate Profile

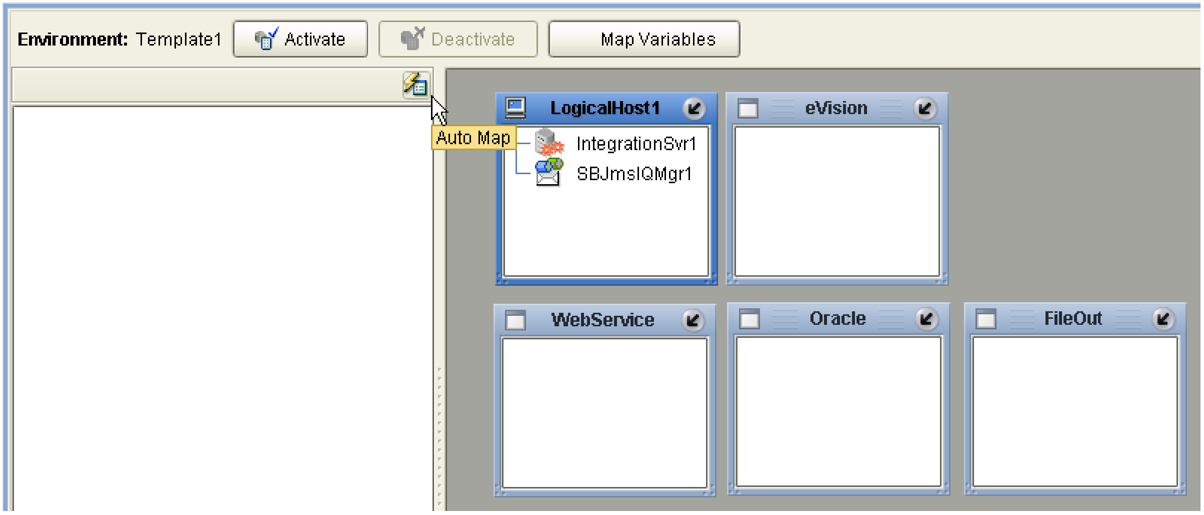


Environment: SunRFIDSuiteJMSSQLServerTemplate

Deployment Editor [Deployment1]
Activate
Deactivate
Map Variables
LogicalHost1
IntegrationSvr1
ObservationProcessCollab1
SBJmsIQMgr1
eGateEpcEventQueue -> ObservationProcessCollab1
MySQLServerSystem
SQL
ObservationProcessCollab1 -> Sqlserver1
Deployment1

You may also use the **Auto Map** feature in **Enteprise Designer**’s **Environment Template**, by clicking the **AutoMap** button, to complete your Deployment Profile.

Figure 20 Auto Map Option



Environment: Template1

5 Click **Activate** to start the activation process.

6 Since your Logical Host is not yet running, click **No** to complete the activation without applying the changes to the Logical Host.

3.2.4 Using the SunRFIDSuiteJMSTigerLogicTemplate Project

This Project uses the Sun RFID Event Manager, JMS input, and TigerLogic Output. To use this Project template you will complete the following general steps:

- Run the New RFID Application Wizard
- Create an Environment.
- Configure the JMS Topic.
- Configure the TigerLogic template.
- Create and activate a Deployment Profile.

Create the SunRFIDSuiteJMSTigerLogicTemplate Project template

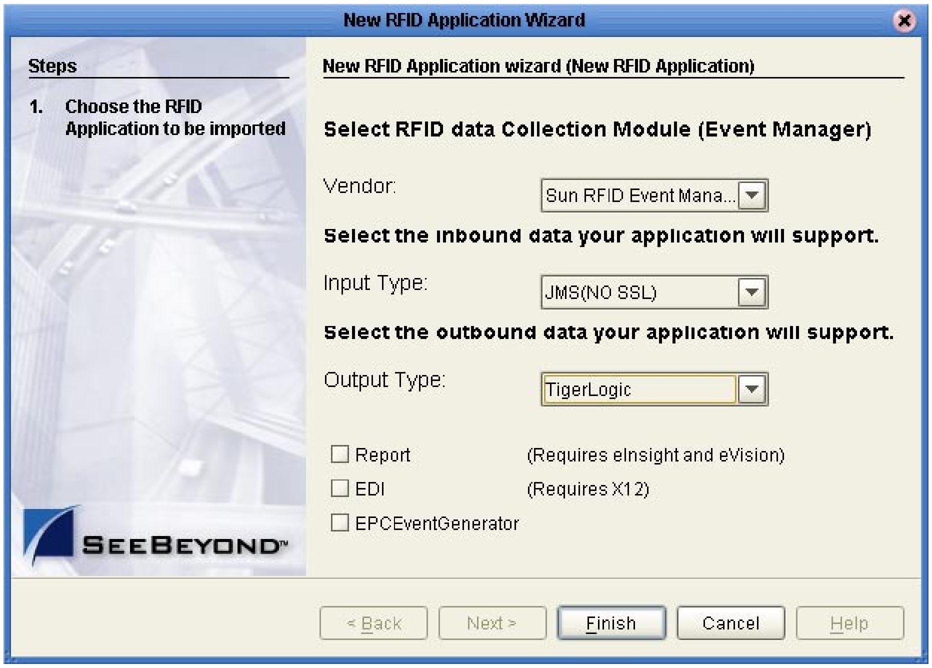
1 Create a new Project in your Repository. This Project will act as the container Project for this RFID application.

2 Right-click the container Project, point to **New**, and then click **New RFID Application**

The New RFID Application Wizard appears.

3 Select the options shown in the following figure, and then click **Finish**.

Figure 21 SunRFIDSuiteJMSTigerLogicTemplate Options


New RFID Application Wizard
Steps
1. Choose the RFID Application to be imported
New RFID Application wizard (New RFID Application)
SEEBEYOND
< Back
Next >
Finish
Cancel
Help

Select RFID data Collection Module (Event Manager)

Vendor: Sun RFID Event Mana...

Select the inbound data your application will support.

Input Type: JMS(NO SSL)

Select the outbound data your application will support.

Output Type: TigerLogic

□ Report (Requires eInsight and eVision)
□ EDI (Requires X12)
□ EPCEventGenerator

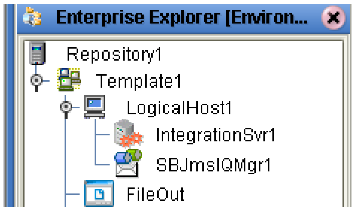
The new RFID Application Wizard creates a new Project template called **SunRFIDSuiteJMSTigerLogicTemplate** in your container Project.

Create and configure the Environment

1. In the Enterprise Explorer pane, click the Environment Explorer tab.
2. Right-click your Repository, and then click **New Environment**.
3. Rename this Environment so it can be easily identified when you create the Deployment Profile (for example, **SunRFIDSuiteJMSTigerLogicTemplate**).
4. Add the following items to your Environment:
   - Logical Host
   - SeeBeyond Integration Server
   - SeeBeyond JMS IQ Manager
   - File External System (outbound)

   Your Environment appears like the following figure.

Figure 22 SunRFIDSuiteJMSTigerLogicTemplate Environment



5. Right-click **FileOut**, and then click **Properties** to view its associated values.

   The file external for your TigerLogic system only acts as a place holder.

Configure the JMS Topic

The JMS Topic requires no configuration. However, the name of the JMS Topic in the Connectivity Map must match the name of the topic to which your Sun RFID Event Manager is publishing. To configure the inbound Topic, you must either rename it in the Connectivity Map or in your Sun RFID system. They both must match.

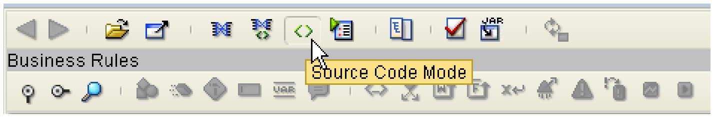
Note: *To rename the Topic in the Connectivity Map, you must first check it out from the Repository. Make sure to check it back in when you are done changing the name.*

Configure the TigerLogic Template

1. In the Enterprise Explorer pane, click the Project Explorer tab.
2. Under the **SunRFIDSuiteJMSTigerLogicTemplate** section, double-click **ObservationProcessCollab**.
3. To check out the collaboration, right-click **ObservationProcessCollab** and select **Check Out**. The Java source code appears in the right pane.

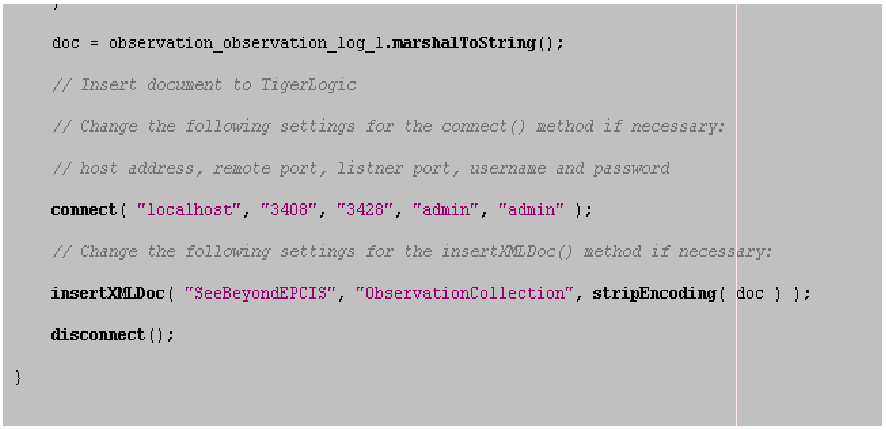
   If the Jave source code does not appear, click the the **Source Code Mode** button from the **Enterprise Designer** toolbar.

Figure 23 Source Code Mode Toolbar Button



4 Navigate the source code until you find the `connect` call.

Figure 24 ObservationProcessCollab Check Out



```
    doc = observation_observation_log_1.marshalToString();

    // Insert document to TigerLogic

    // Change the following settings for the connect() method if necessary:

    // host address, remote port, listner port, username and password

    connect( "localhost", "3408", "3428", "admin", "admin" );

    // Change the following settings for the insertXMLDoc() method if necessary:

    insertXMLDoc( "SeeBeyondEPCIS", "ObservationCollection", stripEncoding( doc ) );

    disconnect();

}
```

5 Modify the parameter values used in the `connect` call if necessary. The call appears as

```
connect( host address, remote port, listner port, username, password
)
```

where

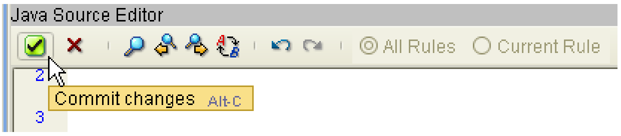
- `host address` is the TCP/IP address of the TigerLogic XML Server
- `remote port` is the port number used by the TigerLogic Server
- `listner port` is the port number of the client listener
- `username` is the login username to the TigerLogic XML Server
- `password` is the login password to the TigerLogic XML Server

Thus, the `connect` call may appear as

```
connect( “localhost”, “3408”, “3428”, “admin”, “admin” )
```

6 Commit your changes by clicking the **Commit Changes** button on the **Java Source Editor** toolbar.

Figure 25 Commit ChangesToolbar Button

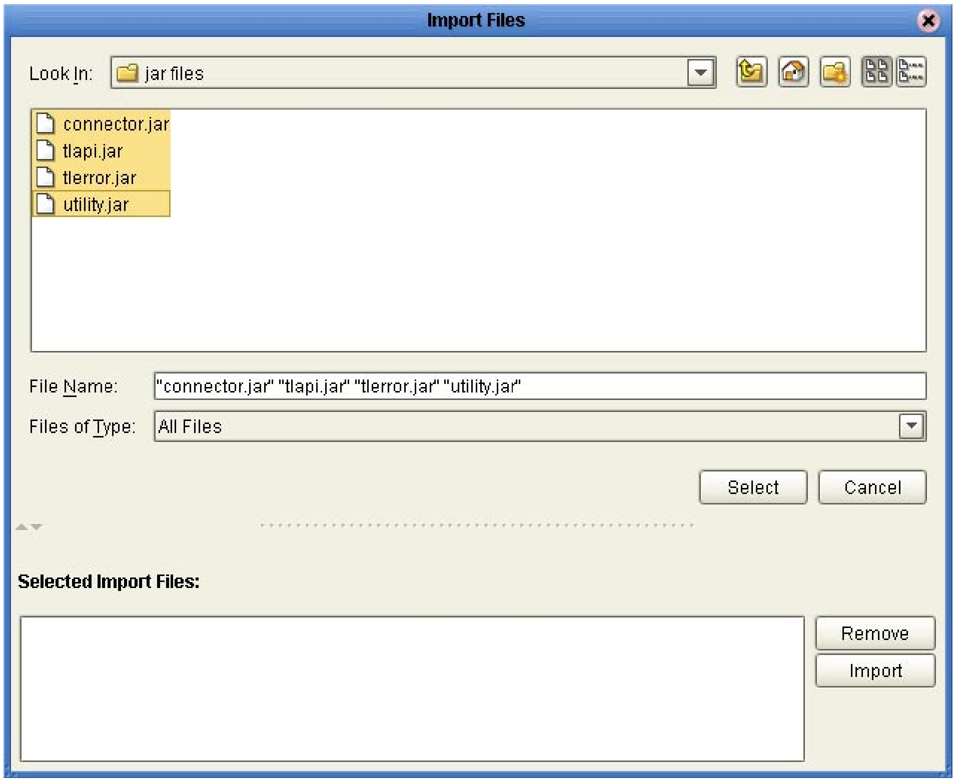


7 Save your changes by clicking the **Save** button on the **Enterprise Designer** toolbar.

8 To add your TigerLogic library to the **ObservationProcessCollab**, right-click the **SunRFIDSuiteJMSTigerLogicTemplate** section, point to **New**, and then click **File**.

The **Import Files** window appears.

Figure 26 Importing TigerLogic Library Files



9 Locate the following TigerLogic jar files in the TigerLogic XML Server directory:

- **connector.jar**
- **tlapi.jar**
- **tlerror.jar**
- **utility.jar**

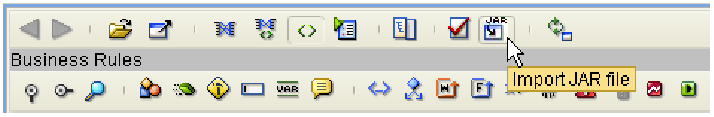
10 Highlight all four files simultaneously by holding the **Ctrl** key and clicking on each file name, click **Select,** and click **Import**.

The four jar files appear in the left pane under the **SunRFIDSuiteJMSTigerLogicTemplate** section.

11 Double-click **ObservationProcessCollab** once again.

12 Click the **Import JAR file** button from the **Enterprise Designer** toolbar.

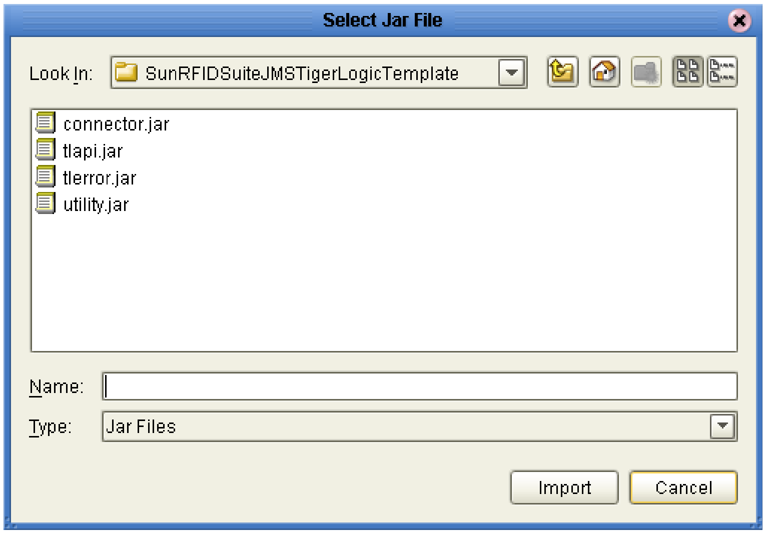
Figure 27 Import JAR File Toolbar Button



13 In the **Add/Remove Jar Files** window, click **Add.**

14 In the **Select Jar File** window, locate the above-mentioned jar files in the folder of your **SunRFIDSuiteJMSTigerLogicTemplate**.

Figure 28 Selecting Jar Files



Look In: SunRFIDSuiteJMSTigerLogicTemplate

connector.jar
tlapi.jar
tlerror.jar
utility.jar

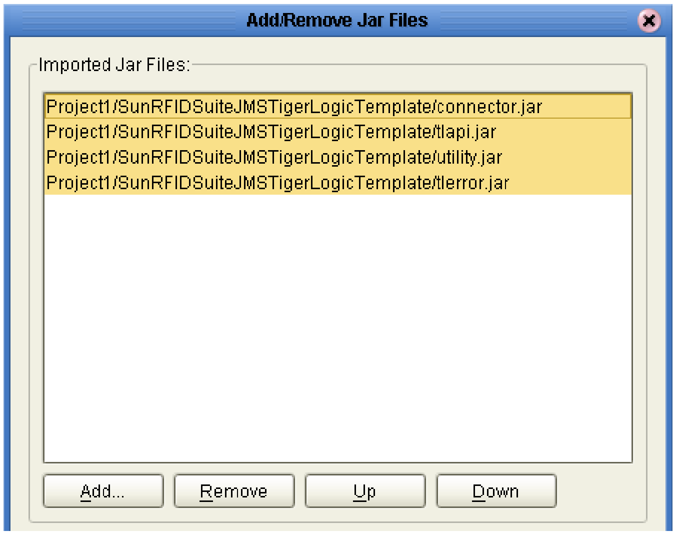
Name:

Type: Jar Files

15 Select each filename and click **Import**.

The **Add/Remove Jar Files** window will be populated with all four file names.

Figure 29 Selecting Jar Files



Imported Jar Files:

Project1/SunRFIDSuiteJMSTigerLogicTemplate/connector.jar
Project1/SunRFIDSuiteJMSTigerLogicTemplate/tlapi.jar
Project1/SunRFIDSuiteJMSTigerLogicTemplate/utility.jar
Project1/SunRFIDSuiteJMSTigerLogicTemplate/tlerror.jar

16 Select all four files, as in the figure above, and click **Add**.

17 Save your changes by clicking the **Save** button on the **Enterprise Designer** toolbar.

Create Deployment Profile

1 In the Enterprise Explorer pane, click the Project Explorer tab.

2 Right-click the **SunRFIDSuiteJMSTigerLogicTemplate** Project, point to **New**, and then click **Deployment Profile**.

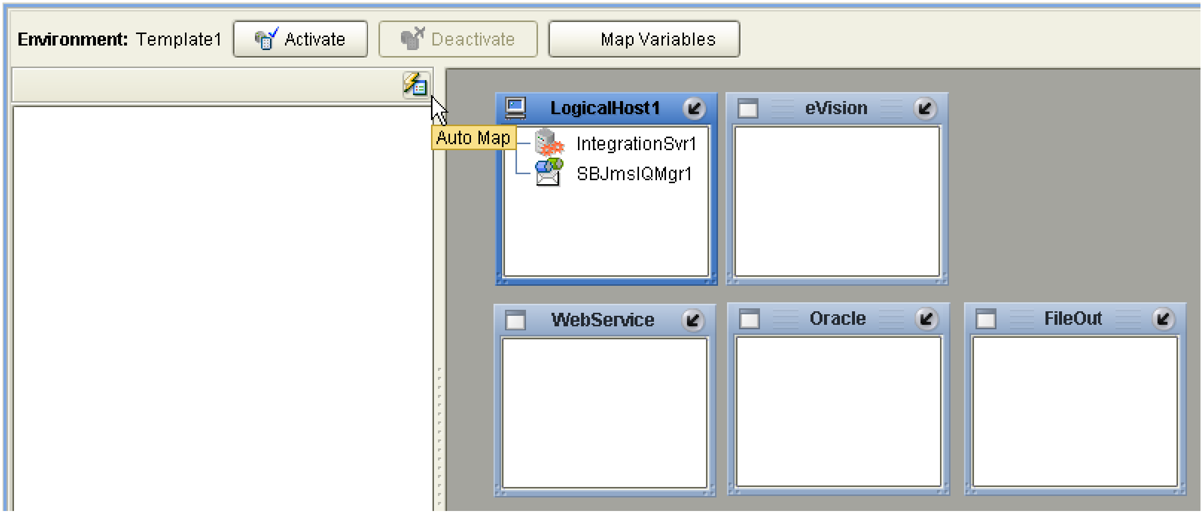
3 Type a **Deployment Profile Name**, select the appropriate **Environment**, and then click **OK**.

Make sure to select the Environment that you created in steps 2 and 3 of the previous procedure.

4 In the Deployment Profile, drag the components to the appropriate location, similar to the procedures for Oracle, DB2, and SQL Server Deployment Profiles.

You may also use the Auto Map feature in Enteprise Designer's Environment Template, by clicking the AutoMap button, to complete your Deployment Profile.

Figure 30 Auto Map Option



5 Click **Activate** to start the activation process.

6 Since your Logical Host is not yet running, click **No** to complete the activation without applying the changes to the Logical Host.

3.3 ConnecTerra Projects

The following sections provide information for using the ConnecTerra Project templates. These topics include:

- **"Using the ConnecterraJMSOracleTemplate Project" on page 50**
- **"Using the ConnecterraJMSDB2Template Project" on page 53**
- **"Using the ConnecterraJMSSQLServerTemplate Project" on page 56**
- **"Using the ConnecterraJMSTigerLogicTemplate Project" on page 59**
- **"Using the HTTPJMSQueueTemplate Project" on page 65**
- **"Using the ConnecterraHTTPOracleTemplate Project" on page 69**
- **"Using the ConnecterraHTTPDB2Template Project" on page 72**
- **"Using the ConnecterraHTTPSQLServerTemplate Project" on page 75**
- **"Using the ConnecterraHTTPTigerLogicTemplate Project" on page 78**

3.3.1 Using the ConnecterraJMSOracleTemplate Project

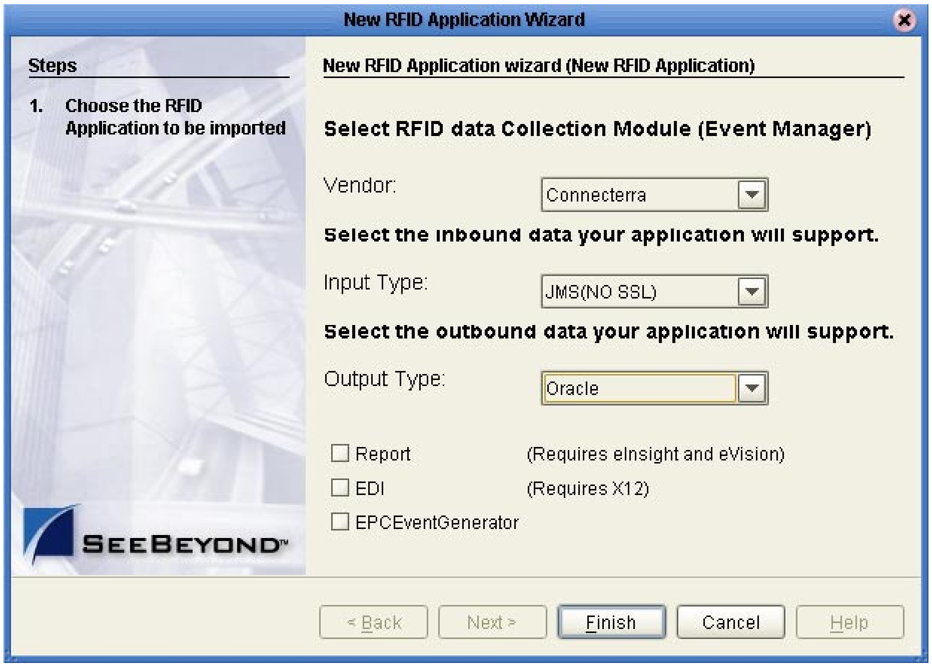
This Project uses ConnecTerra, JMS input, and Oracle Output. To use this Project template you will complete the following general steps:

- Run the New RFID Application Wizard
- Create an Environment.
- Configure the Oracle Server connection.
- Configure the JMS Topic.
- Create and activate a Deployment Profile.

Create the ConnecterraJMSOracleTemplate Project template

1. Create a new Project in your Repository. This Project will act as the container Project for this RFID application.
2. Right-click the container Project, point to **New**, and then click **New RFID Application**.
3. Select the options shown in the following figure, and then click **Finish**.

Figure 31 ConnecterraJMSOracleTemplate Options



New RFID Application Wizard

Steps

1. Choose the RFID Application to be imported

New RFID Application wizard (New RFID Application)

Select RFID data Collection Module (Event Manager)

Vendor: Connecterra

Select the inbound data your application will support.

Input Type: JMS(NO SSL)

Select the outbound data your application will support.

Output Type: Oracle

☐ Report (Requires eInsight and eVision)

☐ EDI (Requires X12)

☐ EPCEventGenerator

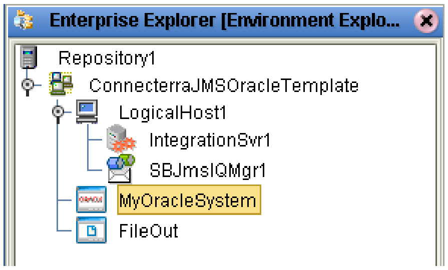
The New RFID Application Wizard creates a new Project template called **ConnecterraJMSOracleTemplate** in your container Project.

Create and configure the Environment

1. In the Enterprise Explorer pane, click the Environment Explorer tab.
2. Right-click your Repository, and then click **New Environment**.
3. Rename this Environment so it can be easily identified when you create the Deployment Profile (for example, **ConnecterraJMSOracleTemplate**).
4. Add the following items to your Environment:
   - Logical Host
   - SeeBeyond Integration Server
   - SeeBeyond JMS IQ Manager
   - Oracle External System (outbound)
   - File External System (outbound)

   Your Environment looks like the following figure.

Figure 32 ConnecterraJMSOracleTemplate Environment



5. Right-click **MyOracleSystem**, and then click **Properties**.
6. Configure the Oracle external as appropriate for your Oracle system, and then click **OK** when finished.

   For complete details on configuring the outbound Oracle System, see the *Oracle eWay Intelligent Adapter User's Guide*.

Configure the JMS Topic

The JMS Topic requires no configuration. However, the name of the JMS Topic in the Connectivity Map must match the name of the topic to which your ConnecTerra Event Manager is publishing. To configure the inbound Topic, you must either rename it in the Connectivity Map or in your ConnecTerra system. They both must match.

Note: *To rename the Topic in the Connectivity Map, you must first check it out from the Repository. Make sure to check it back in when you are done changing the name.*

Create Deployment Profile

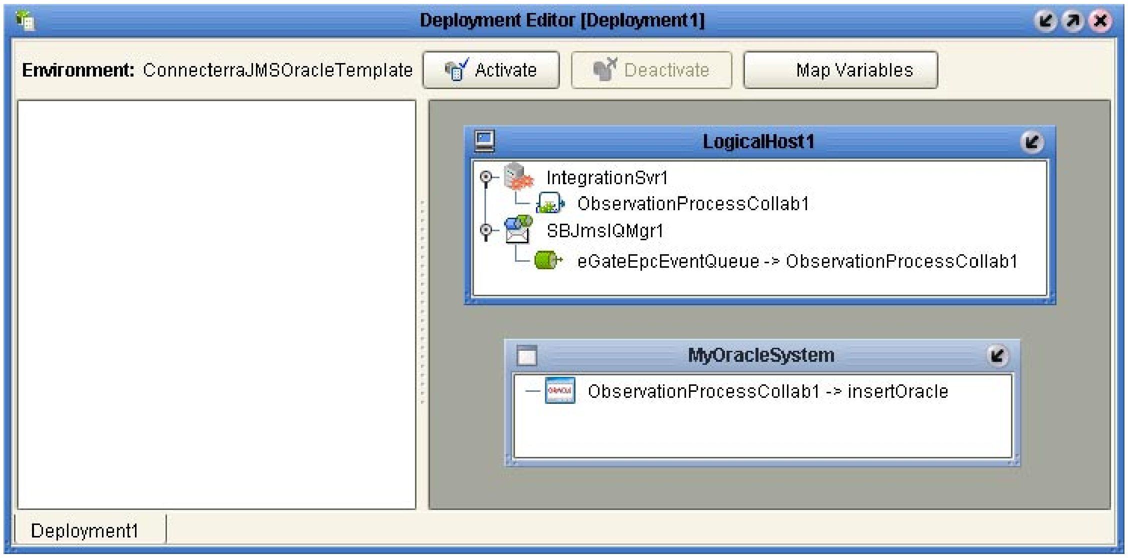
1. In the Enterprise Explorer pane, click the Project Explorer tab.
2. Right-click the **ConnecterraJMSOracleTemplate** Project, point to **New**, and then click **Deployment Profile**.

3 Type a **Deployment Profile Name**, select the appropriate **Environment**, and then click **OK**.

Make sure to select the Environment that you created in steps 2 and 2 of the previous procedure.

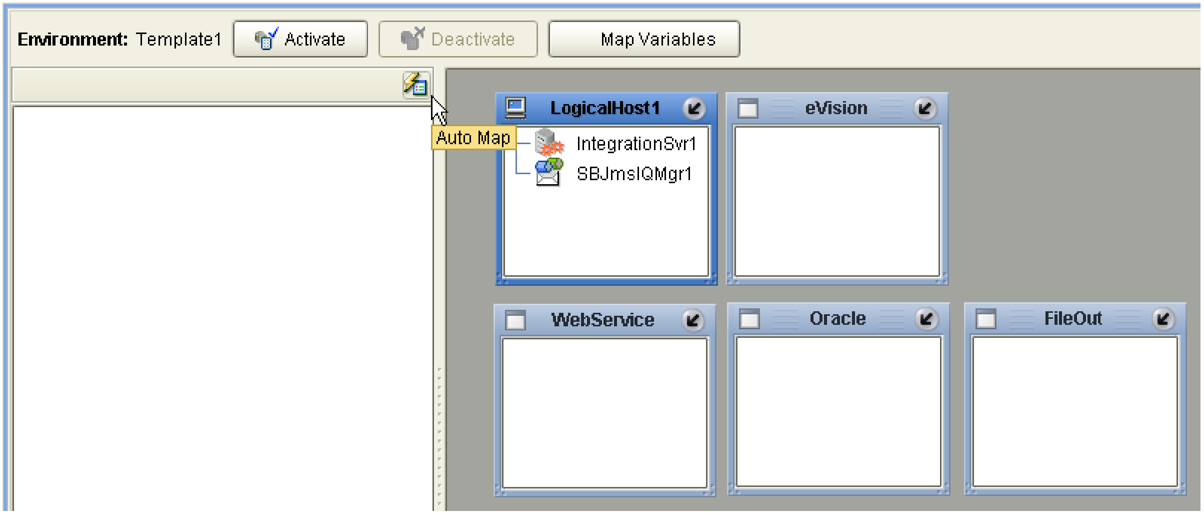
4 In the Deployment Profile, drag the components to the appropriate location as shown in the following figure.

Figure 33 Completed Deployment Profile


Deployment Editor [Deployment1]
Environment: ConnecterraJMSOracleTemplate
Activate
Deactivate
Map Variables
LogicalHost1
IntegrationSvr1
ObservationProcessCollab1
SBJmsIQMgr1
eGateEpcEventQueue -> ObservationProcessCollab1
MyOracleSystem
ObservationProcessCollab1 -> insertOracle
Deployment1

You may also use the **Auto Map** feature in **Enteprise Designer**'s **Environment Template**, by clicking the **AutoMap** button, to complete your Deployment Profile.

Figure 34 Auto Map Option



5 Click **Activate** to start the activation process.

6 Since your Logical Host is not yet running, click **No** to complete the activation without applying the changes to the Logical Host.

3.3.2 Using the ConnecterraJMSDB2Template Project

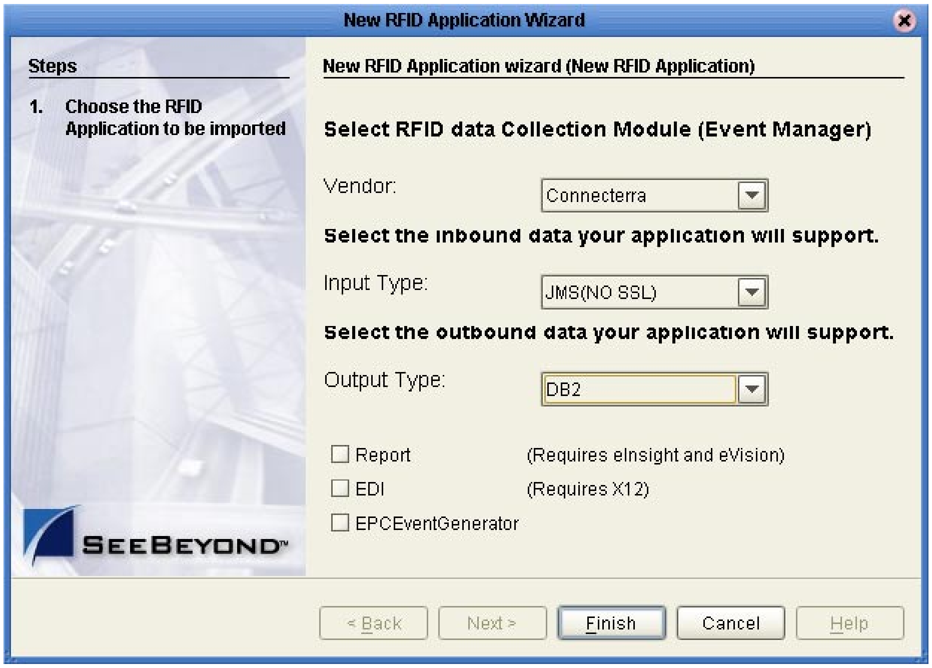
This Project uses ConnecTerra, JMS input, and DB2 UDB Output. To use this Project template you will complete the following general steps:

- Run the New RFID Application Wizard
- Create an Environment.
- Configure the DB2 connection.
- Configure the JMS Topic.
- Create and activate a Deployment Profile.

Create the ConnecterraJMSOracleTemplate Project template

1. Create a new Project in your Repository. This Project will act as the container Project for this RFID application.
2. Right-click the container Project, point to **New**, and then click **New RFID Application**.
3. Select the options shown in the following figure, and then click **Finish**.

Figure 35 ConnecterraJMSDB2Template Options



New RFID Application Wizard

Steps

1. Choose the RFID Application to be imported

New RFID Application wizard (New RFID Application)

Select RFID data Collection Module (Event Manager)

Vendor: Connecterra

Select the inbound data your application will support.

Input Type: JMS(NO SSL)

Select the outbound data your application will support.

Output Type: DB2

□ Report (Requires eInsight and eVision)

□ EDI (Requires X12)

□ EPCEventGenerator

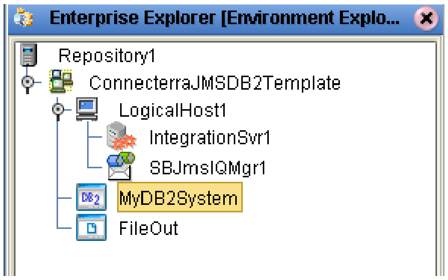
The New RFID Application Wizard creates a new Project template called **ConnecterraJMSDB2Template** in your container Project.

Create and configure the Environment

1. In the Enterprise Explorer pane, click the Environment Explorer tab.
2. Right-click your Repository, and then click **New Environment**.
3. Rename this Environment so it can be easily identified when you create the Deployment Profile (for example, **ConnecterraJMSDB2Template**).
4. Add the following items to your Environment:
   - Logical Host
   - SeeBeyond Integration Server
   - SeeBeyond JMS IQ Manager
   - DB2 External System (outbound)
   - File External System (outbound)

   Your Environment looks like the following figure.

Figure 36 ConnecterraJMSDB2Template Environment



Enterprise Explorer [Environment Explo...

```
Repository1
ConnecterraJMSDB2Template
  LogicalHost1
    IntegrationSvr1
    SBJmsIQMgr1
  MyDB2System
  FileOut
```

5. Right-click **MyDB2System**, and then click **Properties**.
6. Configure the DB2 external as appropriate for your DB2 system, and then click **OK** when finished.

   For complete details on configuring the outbound DB2 System, see the *DB2 Universal Database eWay Intelligent Adapter User's Guide*.

Configure the JMS Topic

The JMS Topic requires no configuration. However, the name of the JMS Topic in the Connectivity Map must match the name of the topic to which your ConnecTerra Event Manager is publishing. To configure the inbound Topic, you must either rename it in the Connectivity Map or in your ConnecTerra system. They both must match.

Note: *To rename the Topic in the Connectivity Map, you must first check it out from the Repository. Make sure to check it back in when you are done changing the name.*

Create Deployment Profile

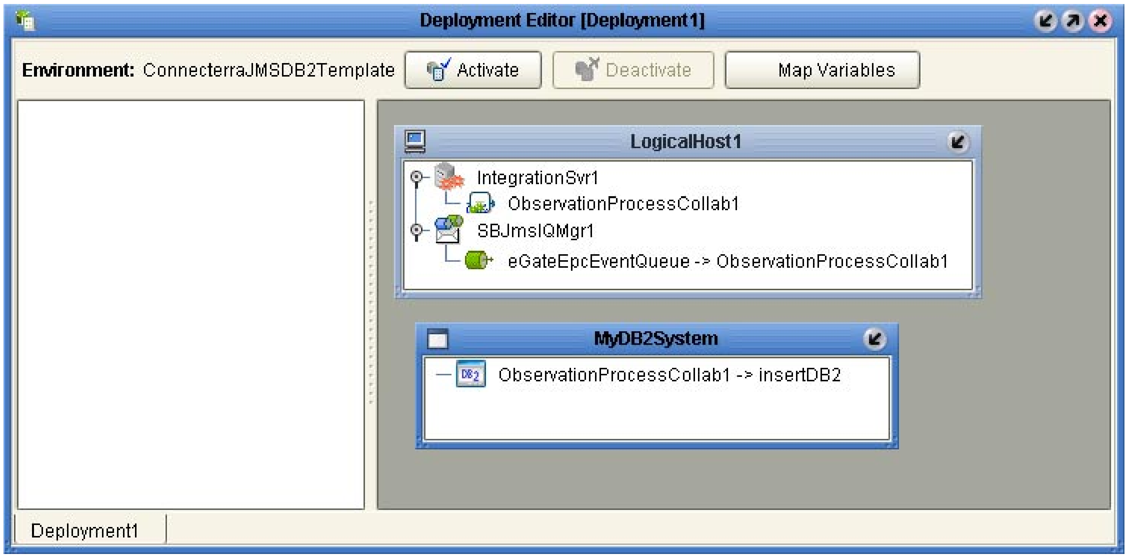
1. In the Enterprise Explorer pane, click the Project Explorer tab.
2. Right-click the **ConnecterraJMSDB2Template** Project, point to **New**, and then click **Deployment Profile**.

3 Type a **Deployment Profile Name**, select the appropriate **Environment**, and then click **OK**.

Make sure to select the Environment that you created in steps 2 and 3 of the previous procedure.

4 In the Deployment Profile, drag the components to the appropriate location as shown in the following figure.

Figure 37 Completed Deployment Profile

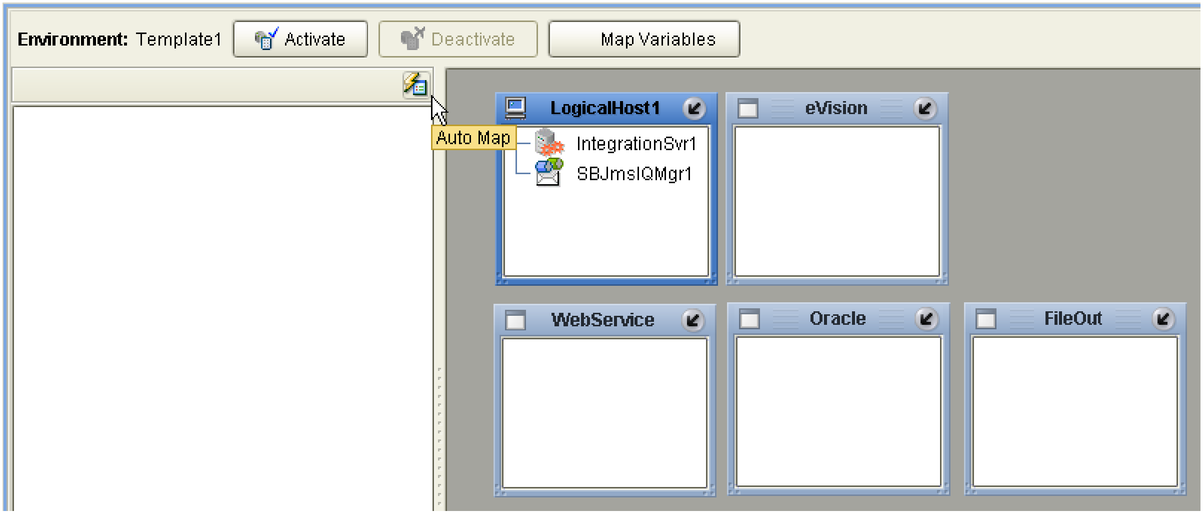


Environment: ConnecterraJMSDB2Template

Deployment Editor [Deployment1]
Activate
Deactivate
Map Variables
LogicalHost1
IntegrationSvr1
ObservationProcessCollab1
SBJmsIQMgr1
eGateEpcEventQueue -> ObservationProcessCollab1
MyDB2System
ObservationProcessCollab1 -> insertDB2
Deployment1

You may also use the **Auto Map** feature in **Enteprise Designer**'s **Environment Template**, by clicking the **AutoMap** button, to complete your Deployment Profile.

Figure 38 Auto Map Option



Environment: Template1

5 Click **Activate** to start the activation process.

6 Since your Logical Host is not yet running, click **No** to complete the activation without applying the changes to the Logical Host.

3.3.3 Using the ConnecterraJMSSQLServerTemplate Project

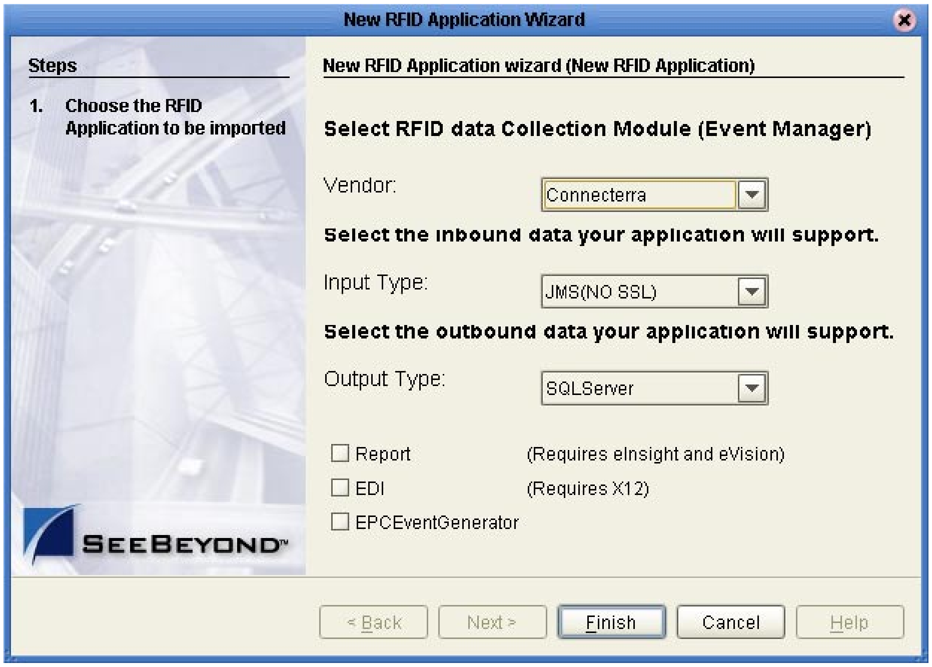
This Project uses ConnecTerra, JMS input, and SQL Server Output. To use this Project template you will complete the following general steps:

- Run the New RFID Application Wizard
- Create an Environment.
- Configure the SQL Server connection.
- Configure the JMS Topic.
- Create and activate a Deployment Profile.

Create the ConnecterraJMSSQLServerTemplate Project template

1. Create a new Project in your Repository. This Project will act as the container Project for this RFID application.
2. Right-click the container Project, point to **New**, and then click **New RFID Application**.
3. Select the options shown in the following figure, and then click **Finish**.

Figure 39 ConnecterraJMSSQLServerTemplate Options



New RFID Application Wizard

Steps

1. Choose the RFID Application to be imported

New RFID Application wizard (New RFID Application)

Select RFID data Collection Module (Event Manager)

Vendor: Connecterra

Select the inbound data your application will support.

Input Type: JMS(NO SSL)

Select the outbound data your application will support.

Output Type: SQLServer

☐ Report (Requires eInsight and eVision)

☐ EDI (Requires X12)

☐ EPCEventGenerator

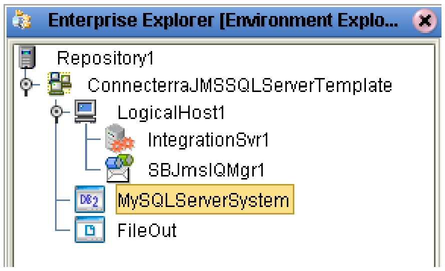
The New RFID Application Wizard creates a new Project template called **ConnecterraJMSSQLServerTemplate** in your container Project.

Create and configure the Environment

1. In the Enterprise Explorer pane, click the Environment Explorer tab.
2. Right-click your Repository, and then click **New Environment**.
3. Rename this Environment so it can be easily identified when you create the Deployment Profile (for example, **ConnecterraJMSSQLServerTemplate**).
4. Add the following items to your Environment:
   - Logical Host
   - SeeBeyond Integration Server
   - SeeBeyond JMS IQ Manager
   - SQL Server External System (outbound)
   - File External System (outbound)

   Your Environment looks like the following figure.

Figure 40 ConnecterraJMSSQLServerTemplate Environment



5. Right-click **MySQLServerSystem**, and then click **Properties**.
6. Configure the SQL Server external as appropriate for your SQL Server system, and then click **OK** when finished.

   For complete details on configuring the outbound SQL Server System, see the *SQL Server eWay Intelligent Adapter User's Guide.*

Configure the JMS Topic

The JMS Topic requires no configuration. However, the name of the JMS Topic in the Connectivity Map must match the name of the topic to which your ConnecTerra Event Manager is publishing. To configure the inbound Topic, you must either rename it in the Connectivity Map or in your ConnecTerra system. They both must match.

Note: *To rename the Topic in the Connectivity Map, you must first check it out from the Repository. Make sure to check it back in when you are done changing the name.*

Create Deployment Profile

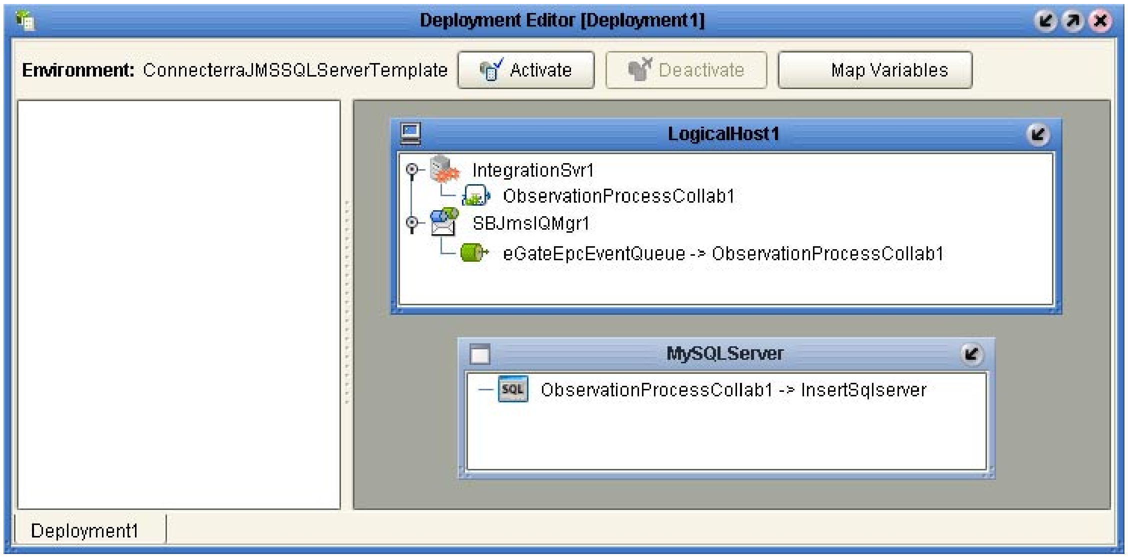
1. In the Enterprise Explorer pane, click the Project Explorer tab.
2. Right-click the **ConnecterraJMSSQLServerTemplate** Project, point to **New**, and then click **Deployment Profile**.

3 Type a **Deployment Profile Name**, select the appropriate **Environment**, and then click **OK**.

Make sure to select the Environment that you created in steps 2 and 3 of the previous procedure.

4 In the Deployment Profile, drag the components to the appropriate location as shown in the following figure.

Figure 41 Completed Deployment Profile

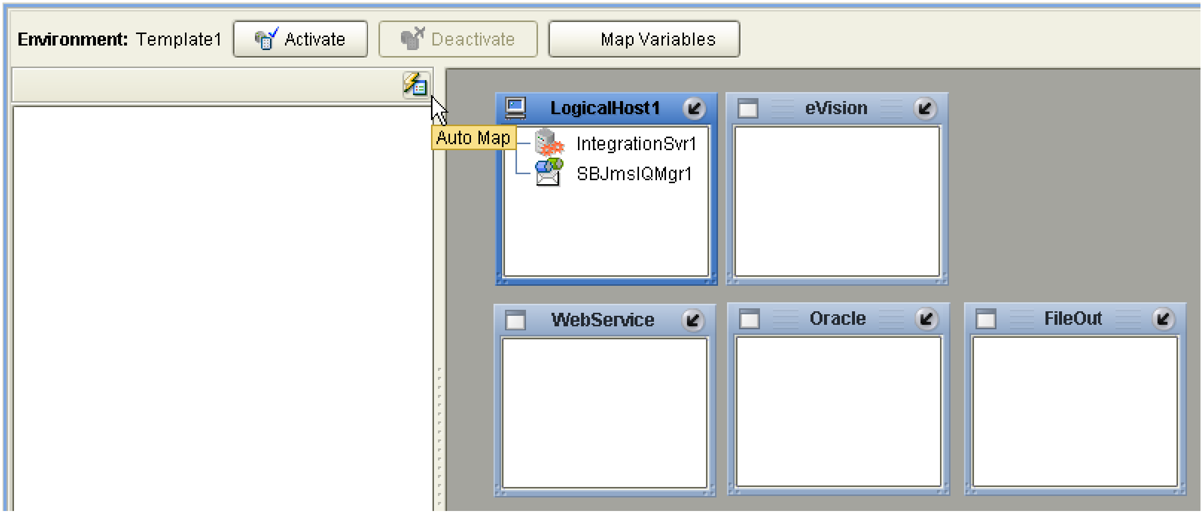


Environment: ConnecterraJMSSQLServerTemplate

Deployment Editor [Deployment1]
Activate
Deactivate
Map Variables
LogicalHost1
IntegrationSvr1
ObservationProcessCollab1
SBJmsIQMgr1
eGateEpcEventQueue -> ObservationProcessCollab1
MySQLServer
SQL
ObservationProcessCollab1 -> InsertSqlserver
Deployment1

You may also use the **Auto Map** feature in **Enteprise Designer**'s **Environment Template**, by clicking the **AutoMap** button, to complete your Deployment Profile.

Figure 42 Auto Map Option



Environment: Template1

5 Click **Activate** to start the activation process.

6 Since your Logical Host is not yet running, click **No** to complete the activation without applying the changes to the Logical Host.

3.3.4 Using the ConnecterraJMSTigerLogicTemplate Project

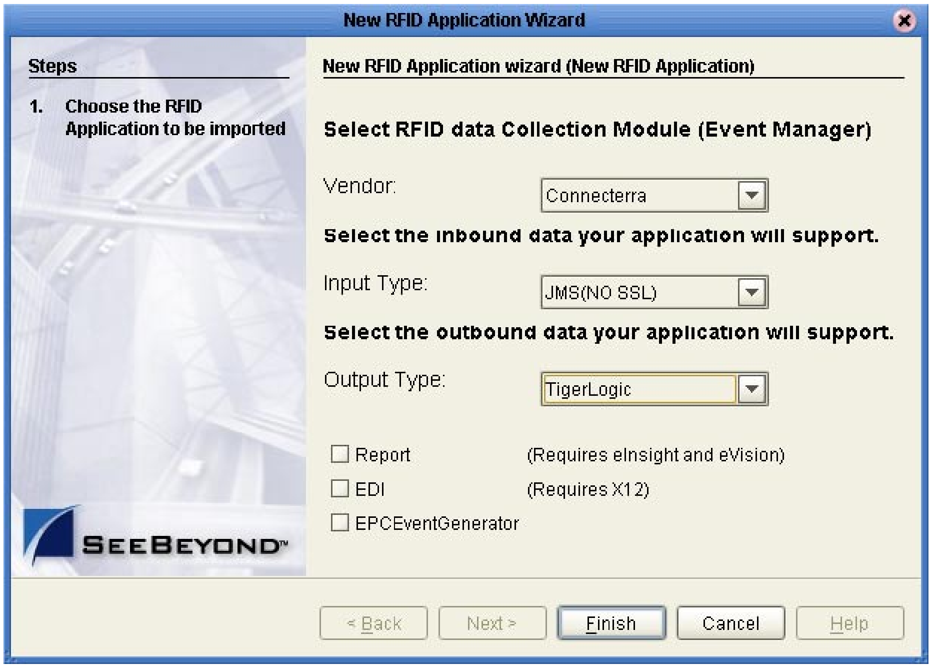
This Project uses ConnecTerra, JMS input, and TigerLogic Output. To use this Project template you will complete the following general steps:

- Run the New RFID Application Wizard
- Create an Environment.
- Configure the JMS Topic.
- Configure the TigerLogic template.
- Create and activate a Deployment Profile.

Create the ConnecterraJMSTigerLogicTemplate Project template

1. Create a new Project in your Repository. This Project will act as the container Project for this RFID application.
2. Right-click the container Project, point to **New**, and then click **New RFID Application**.
3. Select the options shown in the following figure, and then click **Finish**.

Figure 43 ConnecterraJMSTigerLogicTemplate Options


New RFID Application Wizard
Steps
1. Choose the RFID Application to be imported
New RFID Application wizard (New RFID Application)
SEEBEYOND
< Back
Next >
Finish
Cancel
Help

Select RFID data Collection Module (Event Manager)

Vendor: Connecterra

Select the inbound data your application will support.

Input Type: JMS(NO SSL)

Select the outbound data your application will support.

Output Type: TigerLogic

☐ Report (Requires eInsight and eVision)
☐ EDI (Requires X12)
☐ EPCEventGenerator

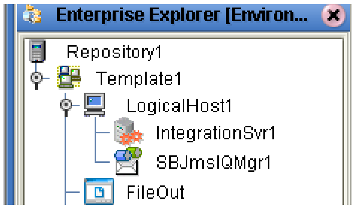
The New RFID Application Wizard creates a new Project template called **ConnecterraJMSTigerLogicTemplate** in your container Project.

Create and configure the Environment

1. In the Enterprise Explorer pane, click the Environment Explorer tab.
2. Right-click your Repository, and then click **New Environment**.
3. Rename this Environment so it can be easily identified when you create the Deployment Profile (for example, **ConnecterraJMSTigerLogicTemplate**).
4. Add the following items to your Environment:
    - Logical Host
    - SeeBeyond Integration Server
    - SeeBeyond JMS IQ Manager
    - File External System (outbound)

   Your Environment looks like the following figure.

Figure 44 ConnecterraJMSTigerLogicTemplate Environment



5. Right-click **FileOut**, and then click **Properties** to view its associated values.

   The file external for your TigerLogic system only acts as a place holder.

Configure the JMS Topic

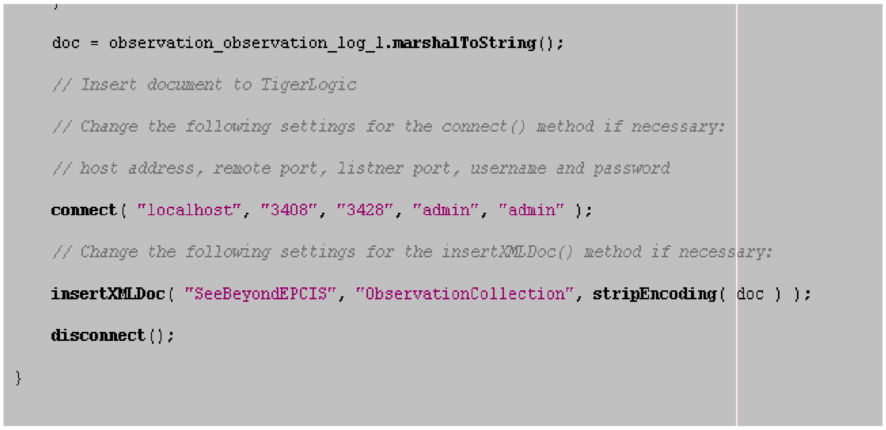
The JMS Topic requires no configuration. However, the name of the JMS Topic in the Connectivity Map must match the name of the topic to which your ConnecTerra Event Manager is publishing. To configure the inbound Topic, you must either rename it in the Connectivity Map or in your ConnecTerra system. They both must match.

Note: *To rename the Topic in the Connectivity Map, you must first check it out from the Repository. Make sure to check it back in when you are done changing the name.*

Configure the TigerLogic Template

1. In the Enterprise Explorer pane, click the Project Explorer tab.
2. Under the **ConnecterraJMSTigerLogicTemplate** section, double-click **ObservationProcessCollab**.
3. To check out the collaboration, right-click **ObservationProcessCollab** and select **Check Out**. The Java source code appears in the right pane.
4. Navigate the source code until you find the `connect` call.

Figure 45 ObservationProcessCollab Check Out



```
    doc = observation_observation_log_1.marshalToString();

    // Insert document to TigerLogic

    // Change the following settings for the connect() method if necessary:

    // host address, remote port, listner port, username and password

    connect( "localhost", "3408", "3428", "admin", "admin" );

    // Change the following settings for the insertXMLDoc() method if necessary:

    insertXMLDoc( "SeeBeyondEPCIS", "ObservationCollection", stripEncoding( doc ) );

    disconnect();

}
```

5 Modify the parameter values used in the `connect` call if necessary. The call appears as

```
connect( host address, remote port, listner port, username, password
)
```

where

- `host address` is the TCP/IP address of the TigerLogic XML Server
- `remote port` is the port number used by the TigerLogic Server
- `listner port` is the port number of the client listener
- `username` is the login username
- `password` is the login password

Thus, the `connect` call may appear as

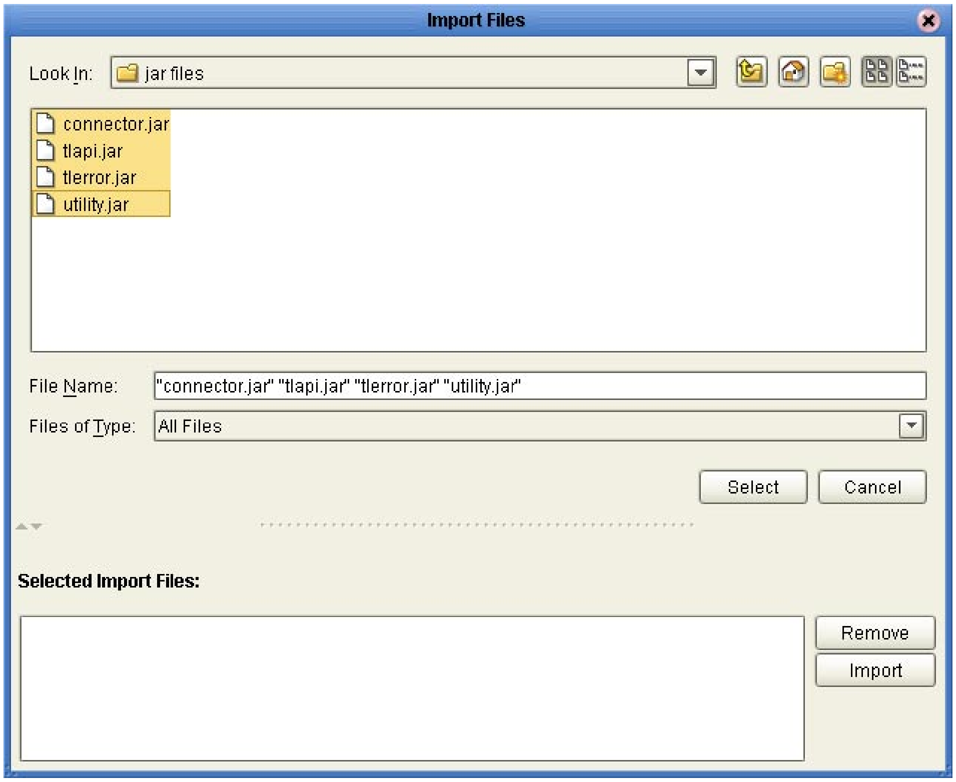
```
connect( “localhost”, “3408”, “3428”, “admin”, “admin” )
```

6 Commit your changes by clicking the **Save** button on the **Enterprise Designer** toolbar.

7 To add your TigerLogic library to the **ObservationProcessCollab**, right-click the **ConnecterraJMSTigerLogicTemplate** section, point to **New**, and then click **File**.

The **Import Files** window appears.

Figure 46 Importing TigerLogic Library Files



Look In: jar files

connector.jar
tlapi.jar
tlerror.jar
utility.jar

File Name: "connector.jar" "tlapi.jar" "tlerror.jar" "utility.jar"

Files of Type: All Files

8 Locate the following TigerLogic jar files:

- `connector.jar`
- `tlapi.jar`
- `utility.jar`

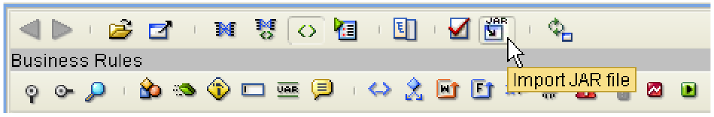
9 Highlight all three files simultaneously by holding the **Ctrl** key and clicking on each file name, click **Select,** and click **Import**.

The three jar files appear in the left pane under the **ConnecterraJMSTigerLogicTemplate** section.

10 Double-click **ObservationProcessCollab** once again.

11 Click the **Import JAR file** button from the toolbar.

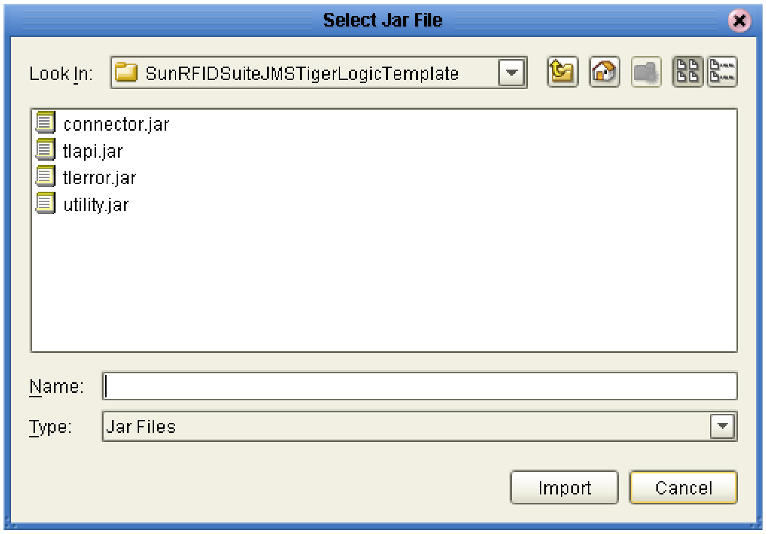
Figure 47 Import JAR File Button



12 In the **Add/Remove Jar Files** window, click **Add.**

13 In the **Select Jar File** window, locate the above-mentioned jar files in the folder of your **ConnecterraJMSTigerLogicTemplate**.

Figure 48 Selecting Jar Files



Select Jar File

Look In: SunRFIDSuiteJMSTigerLogicTemplate

connector.jar
tlapi.jar
tlerror.jar
utility.jar

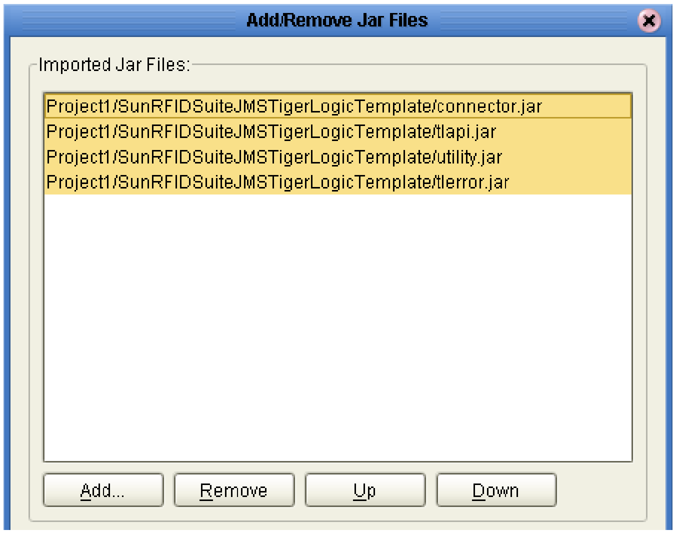
Name:

Type: Jar Files

14 Select each filename and click **Import**.

The **Add/Remove Jar Files** window should be populated with all three file names.

Figure 49 Selecting Jar Files



Add/Remove Jar Files

Imported Jar Files:

Project1/SunRFIDSuiteJMSTigerLogicTemplate/connector.jar
Project1/SunRFIDSuiteJMSTigerLogicTemplate/tlapi.jar
Project1/SunRFIDSuiteJMSTigerLogicTemplate/utility.jar
Project1/SunRFIDSuiteJMSTigerLogicTemplate/tlerror.jar

15 Select all three files, as in the figure above, and click **Add**.

16 Commit your changes by clicking the **Save** button on the **Enterprise Designer** toolbar.

Create Deployment Profile

1 In the Enterprise Explorer pane, click the Project Explorer tab.

2 Right-click the **ConnecterraJMSTigerLogicTemplate** Project, point to **New**, and then click **Deployment Profile**.

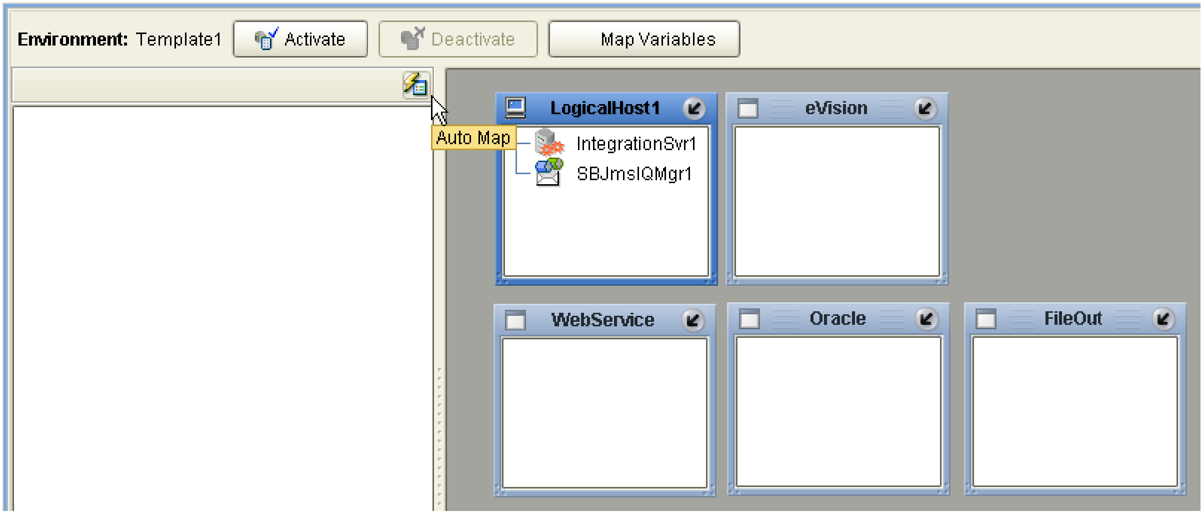
3 Type a **Deployment Profile Name**, select the appropriate **Environment**, and then click **OK**.

Make sure to select the Environment that you created in steps 2 and 3 of the previous procedure.

4 In the Deployment Profile, drag the components to the appropriate location, similar to the procedures for Oracle, DB2, and SQL Server Deployment Profiles.

You may also use the **Auto Map** feature in **Enteprise Designer**'s **Environment Template**, by clicking the **AutoMap** button, to complete your Deployment Profile.

Figure 50 Auto Map Option



5 Click **Activate** to start the activation process.

6 Since your Logical Host is not yet running, click **No** to complete the activation without applying the changes to the Logical Host.

3.3.5 Using the HTTPJMSQueueTemplate Project

This Project uses ConnecTerra, HTTP input, and JMS Output. The output can be to a JMS queue or topic. This section uses a queue for the output. In your implementation, you can use a queue or a topic. The rest of the configuration is the same.

To use this Project template you will complete the following general steps:

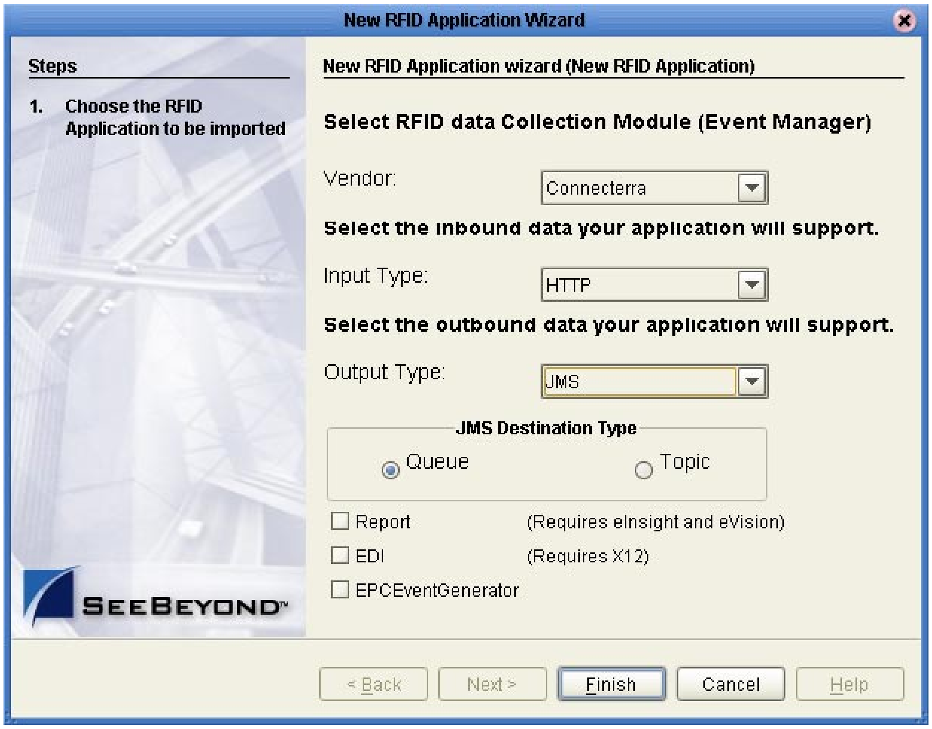
- Run the New RFID Application Wizard
- Create an Environment.
- Configure the HTTP Servlet
- Configure the JMS Queue.
- Create and activate a Deployment Profile.

Create the HTTPJMSQueueTemplate Project template

1. Create a new Project in your Repository. This Project will act as the container Project for this RFID application.
2. Right-click the container Project, point to **New**, and then click **New RFID Application**.
3. Select the options shown in the following figure, and then click **Finish**.

Note: *Select **Queue** or **Topic** depending on your requirements for the JMS output.*

Figure 51 HTTPJMSQueueTemplate Options



New RFID Application Wizard

Steps

1. Choose the RFID Application to be imported

New RFID Application wizard (New RFID Application)

Select RFID data Collection Module (Event Manager)

Vendor: Connecterra

Select the inbound data your application will support.

Input Type: HTTP

Select the outbound data your application will support.

Output Type: JMS

JMS Destination Type

● Queue ○ Topic

□ Report (Requires eInsight and eVision)

□ EDI (Requires X12)

□ EPCEventGenerator

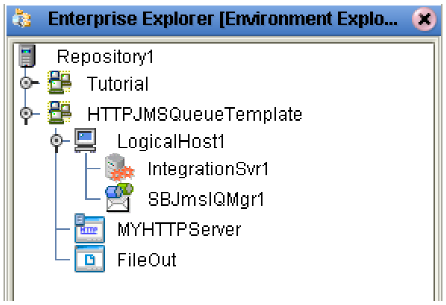
The New RFID Application Wizard creates a new Project template called **HTTPJMSQueueTemplate** in your container Project. (If you selected Topic, then the **HTTPJMSTopicTemplate** appears).

Create and configure the Environment

1. In the Enterprise Explorer pane, click the Environment Explorer tab.
2. Right-click your Repository, and then click **New Environment**.
3. Rename this Environment so it can be easily identified when you create the Deployment Profile (for example, **HTTPJMSQueueTemplate**).
4. Add the following items to your Environment:
   - Logical Host
   - SeeBeyond Integration Server
   - SeeBeyond JMS IQ Manager
   - HTTP Server External System (outbound)
   - File External System (outbound)

   Your Environment looks like the following figure.

Figure 52 HTTPJMSQueueTemplate Environment



Configure the HTTP Servlet

You have the option of using the default Servlet URL for the HTTP server or using a different one.

For complete information on the servlet URL, see the *HTTP(S) eWay Intelligent Adapter User's Guide*.

Configure the JMS Queue or Topic

Depending on whether you selected **Queue** or **Topic** in the New RFID Application Wizard (step 3 in the **procedure on page 65**), your configuration steps will vary.

If you chose to use a Queue, no configuration is necessary. However, the name of the JMS Queue in the Connectivity Map must match the name of the queue to which your JMS subscribers are subscribing. You must either rename the Queue in the Connectivity Map or in your external JMS subscribers. They both must match.

If you chose to use a Topic, you can configure this in the Connectivity Map. For complete information on configuring JMS Topics, see the *eGate Integrator JMS Reference Guide*.

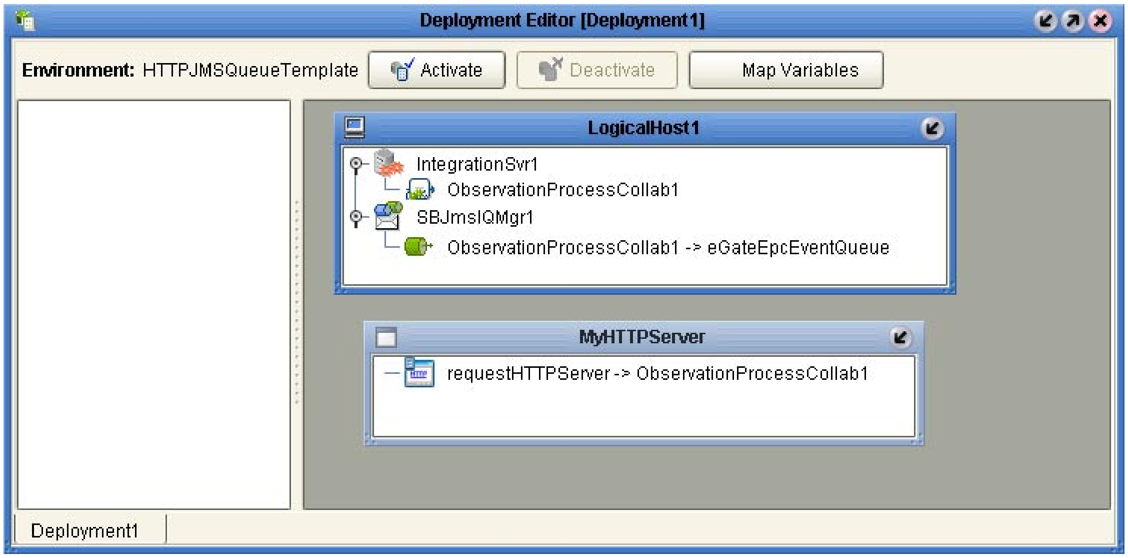
Note: *To modify the Queue or Topic in the Connectivity Map, you must first check it out from the Repository. Make sure to check it back in when you are done making changes.*

Create Deployment Profile

1. In the Enterprise Explorer pane, click the Project Explorer tab.
2. Right-click the **HTTPJMSQueueTemplate** Project, point to **New**, and then click **Deployment Profile**.
3. Type a **Deployment Profile Name**, select the appropriate **Environment**, and then click **OK**.

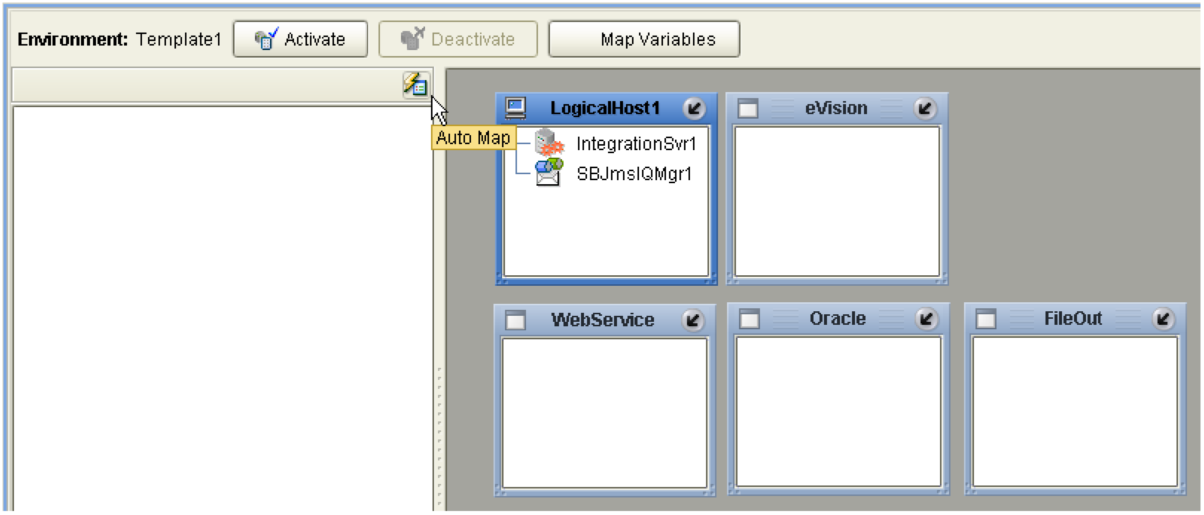
   Make sure to select the Environment that you created in steps 2 and 3 of the previous procedure.
4. In the Deployment Profile, drag the components to the appropriate location as shown in the following figure.

Figure 53 Completed Deployment Profile


Deployment Editor [Deployment1]
Environment: HTTPJMSQueueTemplate
Activate
Deactivate
Map Variables
LogicalHost1
IntegrationSvr1
ObservationProcessCollab1
SBJmsIQMgr1
ObservationProcessCollab1 -> eGateEpcEventQueue
MyHTTPServer
requestHTTPServer -> ObservationProcessCollab1
Deployment1

You may also use the **Auto Map** feature in **Enteprise Designer**’s **Environment Template**, by clicking the **AutoMap** button, to complete your Deployment Profile.

Figure 54 Auto Map Option



5 Click **Activate** to start the activation process.

6 Since your Logical Host is not yet running, click **No** to complete the activation without applying the changes to the Logical Host.

3.3.6 Using the ConnecterraHTTPOracleTemplate Project

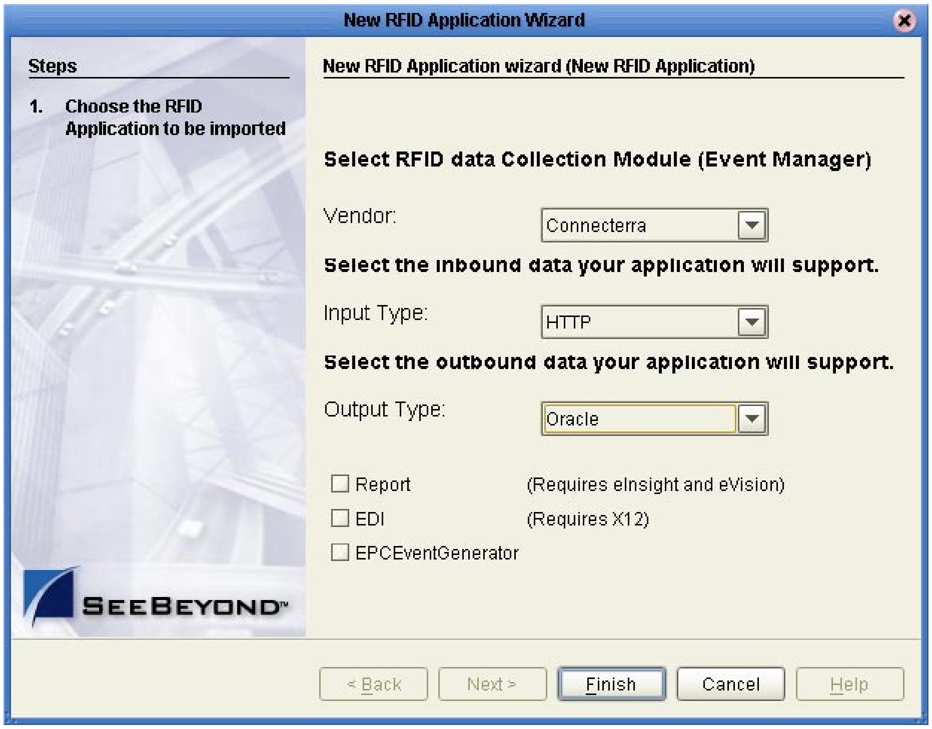
This Project uses ConnecTerra, HTTP input, and Oracle output. To use this Project template you will complete the following general steps:

- Run the New RFID Application Wizard
- Create an Environment.
- Configure the HTTP Servlet
- Configure the Oracle Server connection.
- Create and activate a Deployment Profile.

Create the ConnecterraHTTPOracleTemplate Project template

1. Create a new Project in your Repository. This Project will act as the container Project for this RFID application.
2. Right-click the container Project, point to **New**, and then click **New RFID Application**.
3. Select the options shown in the following figure, and then click **Finish**.

Figure 55 ConnecterraHTTPOracleTemplate Options



New RFID Application Wizard

Steps

1. Choose the RFID Application to be imported

New RFID Application wizard (New RFID Application)

Select RFID data Collection Module (Event Manager)

Vendor: Connecterra

Select the inbound data your application will support.

Input Type: HTTP

Select the outbound data your application will support.

Output Type: Oracle

□ Report (Requires eInsight and eVision)

□ EDI (Requires X12)

□ EPCEventGenerator

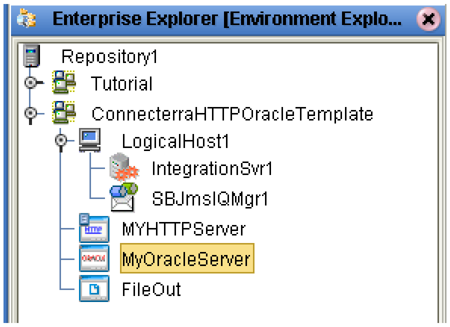
The New RFID Application Wizard creates a new Project template called **ConnecterraHTTPOracleTemplate** in your container Project.

Create and configure the Environment

1. In the Enterprise Explorer pane, click the Environment Explorer tab.
2. Right-click your Repository, and then click **New Environment**.
3. Rename this Environment so it can be easily identified when you create the Deployment Profile (for example, **ConnecterraHTTPOracleTemplate**).
4. Add the following items to your Environment:
   - Logical Host
   - SeeBeyond Integration Server
   - HTTP Server External System
   - Oracle External System (outbound)
   - File External System (outbound)

   Your Environment looks like the following figure.

Figure 56 ConnecterraHTTPOracleTemplate Environment



5. Right-click **MyOracleServer**, and then click **Properties**
6. Configure the Oracle external as appropriate for your Oracle system, and then click **OK** when finished.

   For complete details on configuring the outbound Oracle System, see the *Oracle eWay Intelligent Adapter User's Guide*.

Configure the HTTP Servlet

You have the option of using the default Servlet URL for the HTTP server or using a different one.

For complete information on the servlet URL, see the *HTTP(S) eWay Intelligent Adapter User's Guide*.

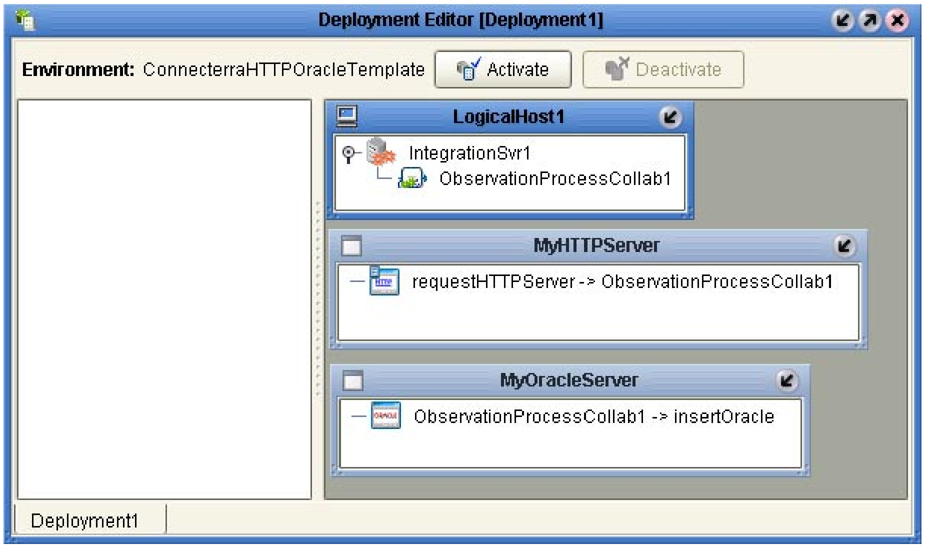
Create Deployment Profile

1. In the Enterprise Explorer pane, click the Project Explorer tab.
2. Right-click the **ConnecterraHTTPOracleTemplate** Project, point to **New**, and then click **Deployment Profile**.
3. Type a **Deployment Profile Name**, select the appropriate **Environment**, and then click **OK**.

Make sure to select the Environment that you created in steps 2 and 3 of the previous procedure.

4 In the Deployment Profile, drag the components to the appropriate location as shown in the following figure.

Figure 57 Completed Deployment Profile

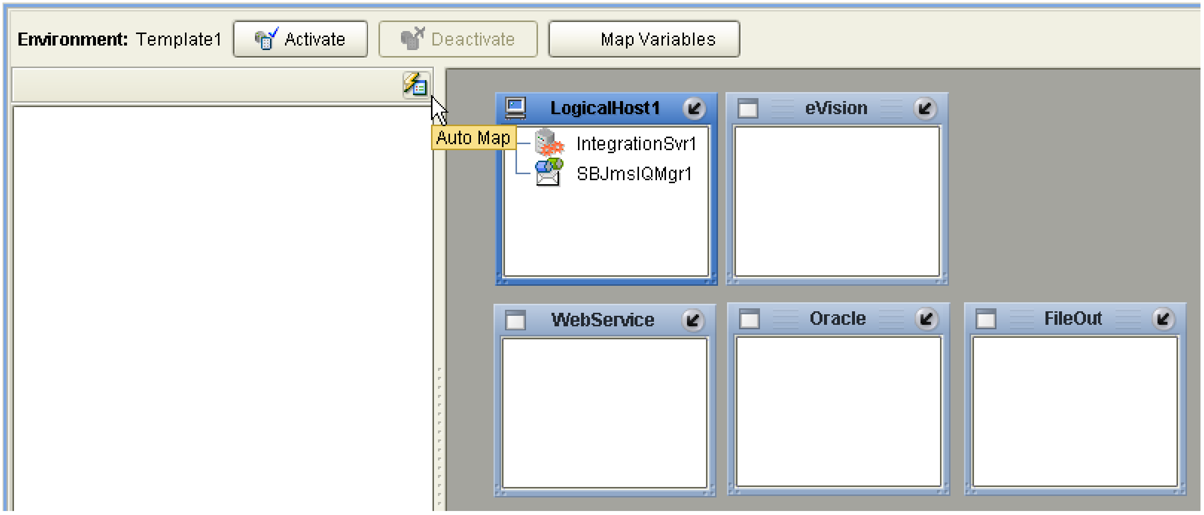


Deployment Editor [Deployment1]
Activate
Deactivate
LogicalHost1
IntegrationSvr1
ObservationProcessCollab1
MyHTTPServer
requestHTTPServer -> ObservationProcessCollab1
MyOracleServer
ObservationProcessCollab1 -> insertOracle
Deployment1

Environment: ConnecterraHTTPOracleTemplate

You may also use the **Auto Map** feature in **Enteprise Designer**'s **Environment Template**, by clicking the **AutoMap** button, to complete your Deployment Profile.

Figure 58 Auto Map Option



Environment: Template1

5 Click **Activate** to start the activation process.

6 Since your Logical Host is not yet running, click **No** to complete the activation without applying the changes to the Logical Host.

3.3.7 Using the ConnecterraHTTPDB2Template Project

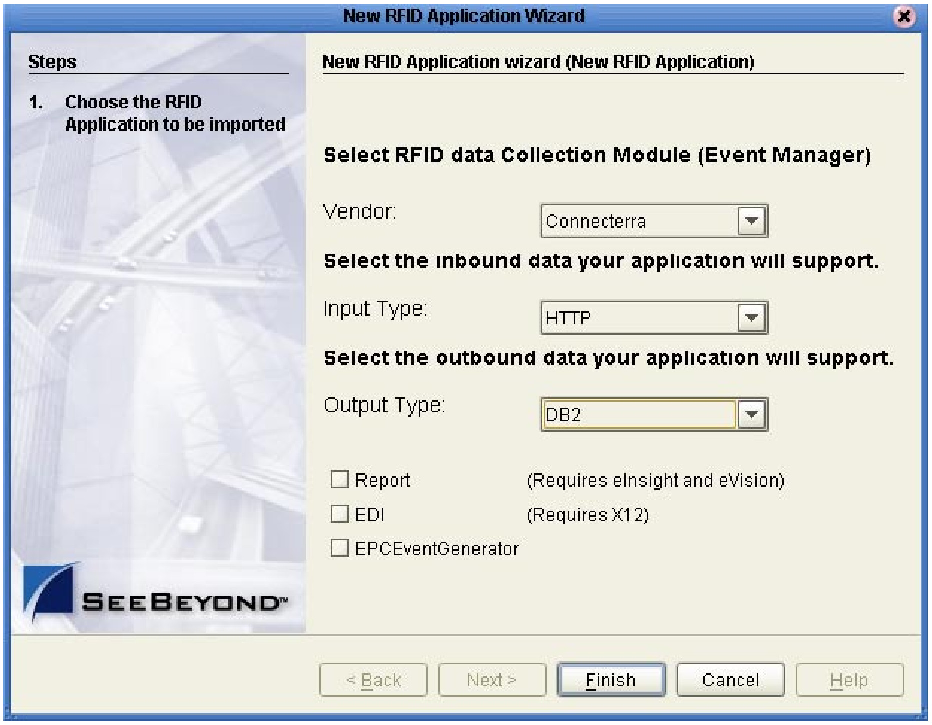
This Project uses ConnecTerra, HTTP input, and DB2 UDB output. To use this Project template you will complete the following general steps:

- Run the New RFID Application Wizard.
- Create an Environment.
- Configure the HTTP Servlet.
- Configure the DB2 connection.
- Create and activate a Deployment Profile.

Create the ConnecterraHTTPDB2Template Project template

1. Create a new Project in your Repository. This Project will act as the container Project for this RFID application.
2. Right-click the container Project, point to **New**, and then click **New RFID Application**.
3. Select the options shown in the following figure, and then click **Finish**.

Figure 59 ConnecterraHTTPDB2Template Options



New RFID Application Wizard

Steps

1. Choose the RFID Application to be imported

New RFID Application wizard (New RFID Application)

Select RFID data Collection Module (Event Manager)

Vendor: Connecterra

Select the inbound data your application will support.

Input Type: HTTP

Select the outbound data your application will support.

Output Type: DB2

□ Report (Requires eInsight and eVision)

□ EDI (Requires X12)

□ EPCEventGenerator

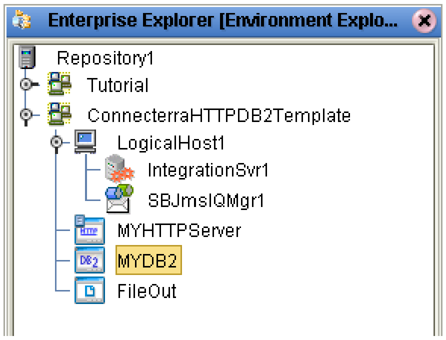
The New RFID Application Wizard creates a new Project template called **ConnecterraHTTPDB2Template** in your container Project.

Create and configure the Environment

1. In the Enterprise Explorer pane, click the Environment Explorer tab.
2. Right-click your Repository, and then click **New Environment**.
3. Rename this Environment so it can be easily identified when you create the Deployment Profile (for example, **ConnecterraHTTPDB2Template**).
4. Add the following items to your Environment:
   - Logical Host
   - SeeBeyond Integration Server
   - HTTP Server External System
   - DB2 External System (outbound)
   - File External System (outbound)

   Your Environment looks like the following figure.

Figure 60 ConnecterraHTTPDB2Template Environment



5. Right-click **MyDB2**, and then click **Properties**.
6. Configure the DB2 external as appropriate for your DB2 system, and then click **OK** when finished.

   For complete details on configuring the outbound DB2 System, see the *DB2 Universal Database eWay Intelligent Adapter User's Guide*.

Configure the HTTP Servlet

You have the option of using the default Servlet URL for the HTTP server or using a different one.

For complete information on the servlet URL, see the *HTTP(S) eWay Intelligent Adapter User's Guide*.

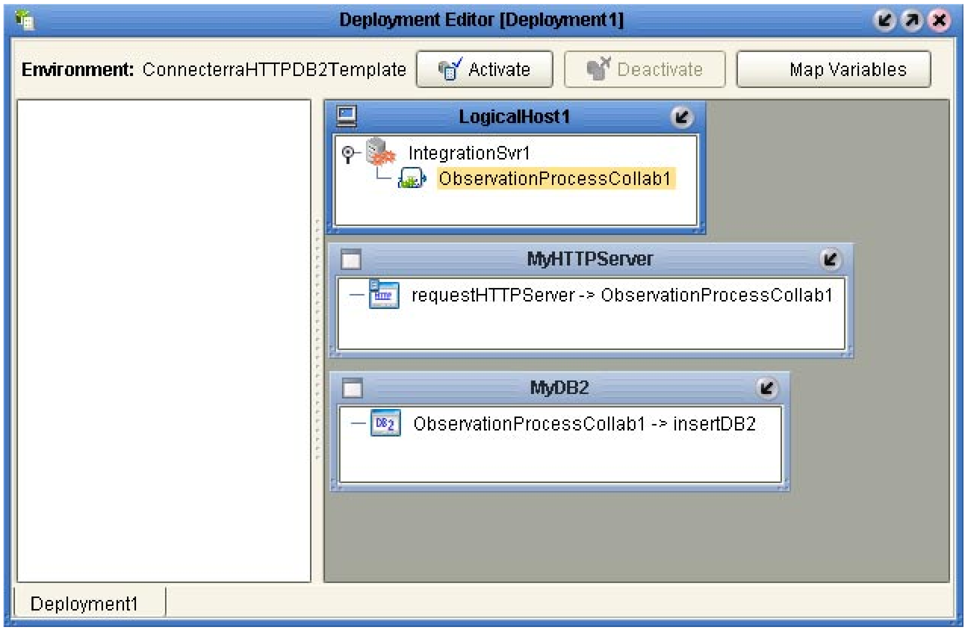
Create the Deployment Profile

1. In the Enterprise Explorer pane, click the Project Explorer tab.
2. Right-click the **ConnecterraHTTPDB2Template** Project, point to **New**, and then click **Deployment Profile**.
3. Type a **Deployment Profile Name**, select the appropriate **Environment**, and then click **OK**.

Make sure to select the Environment that you created in steps 2 and 3 of the previous procedure.

4 In the Deployment Profile, drag the components to the appropriate location as shown in the following figure.

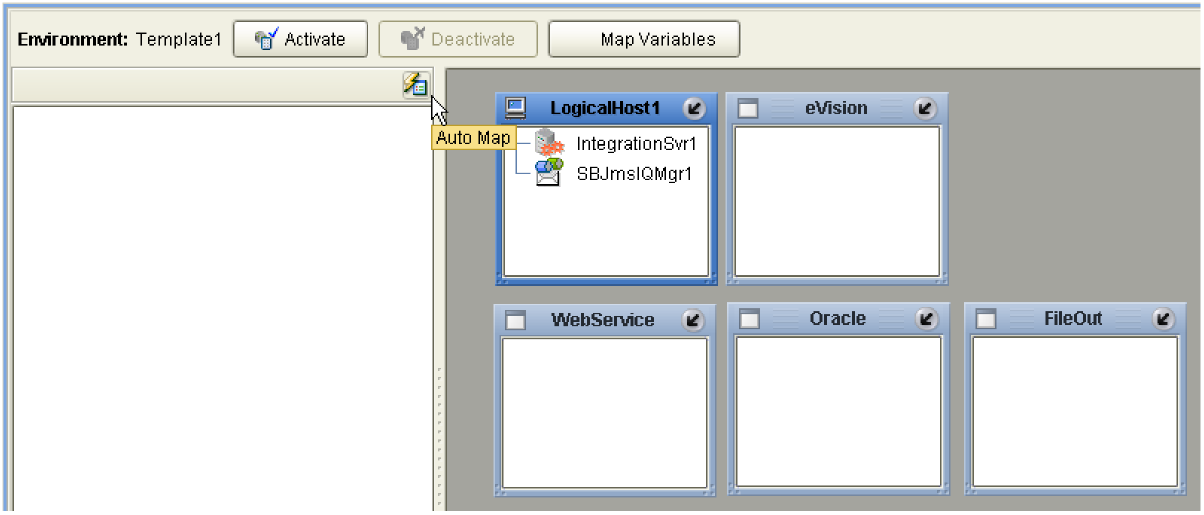
Figure 61 Completed Deployment Profile



Deployment Editor [Deployment1]
Environment: ConnecterraHTTPDB2Template
Activate
Deactivate
Map Variables
LogicalHost1
IntegrationSvr1
ObservationProcessCollab1
MyHTTPServer
requestHTTPServer -> ObservationProcessCollab1
MyDB2
ObservationProcessCollab1 -> insertDB2
Deployment1

You may also use the **Auto Map** feature in **Enteprise Designer**’s **Environment Template**, by clicking the **AutoMap** button, to complete your Deployment Profile.

Figure 62 Auto Map Option



5 Click **Activate** to start the activation process.

6 Since your Logical Host is not yet running, click **No** to complete the activation without applying the changes to the Logical Host.

3.3.8 Using the ConnecterraHTTPSQLServerTemplate Project

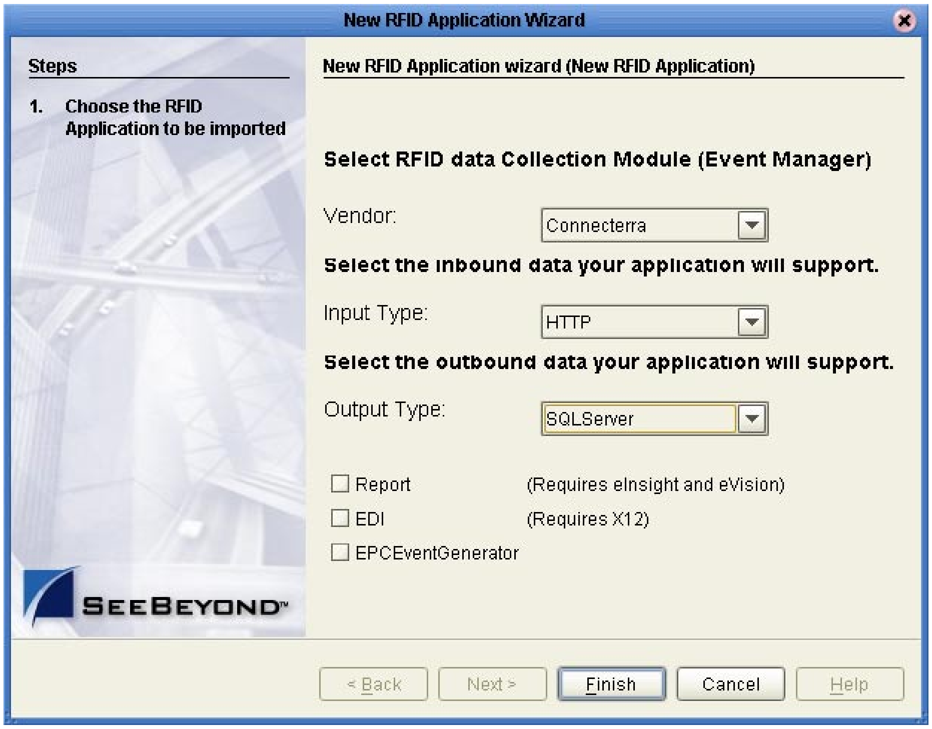
This Project uses ConnecTerra, HTTP input, and SQL Server output. To use this Project template you will complete the following general steps:

- Run the New RFID Application Wizard.
- Create an Environment.
- Configure the HTTP Servlet.
- Configure the SQL Server connection.
- Create and activate a Deployment Profile.

Create the ConnecterraHTTPSQLServerTemplate Project template

1. Create a new Project in your Repository. This Project will act as the container Project for the RFID application.
2. Right-click the container Project, point to **New**, and then click **New RFID Application**.
3. Select the options shown in the following figure, and then click **Finish**.

Figure 63



New RFID Application Wizard

Steps

1. Choose the RFID Application to be imported

New RFID Application wizard (New RFID Application)

Select RFID data Collection Module (Event Manager)

Vendor: Connecterra

Select the inbound data your application will support.

Input Type: HTTP

Select the outbound data your application will support.

Output Type: SQLServer

□ Report (Requires eInsight and eVision)

□ EDI (Requires X12)

□ EPCEventGenerator

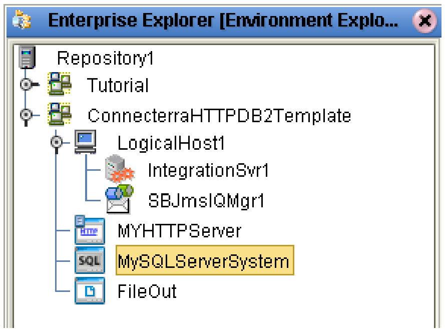
The New RFID Application Wizard creates a new Project template called **ConnecterraHTTPSQLServerTemplate** in your container Project.

Create and configure the Environment

1. In the Enterprise Explorer pane, click the Environment Explorer tab.
2. Right-click your Repository, and then click **New Environment**.
3. Rename this Environment so it can be easily identified when you create the Deployment Profile (for example, **ConnecterraHTTPSQLServerTemplate**).
4. Add the following items to your Environment:
   - Logical Host
   - SeeBeyond Integration Server
   - HTTP Server External System
   - SQL Server External System (outbound)
   - File External System (outbound)

   Your Environment looks like the following figure.

Figure 64 ConnecterraHTTPSQLServerTemplate Environment



5. Right-click **MySQLServerSystem**, and then click **Properties**.
6. Configure the SQL Server external as appropriate to your SQL Server system, and then click **OK** when finished.

   For complete details on configuring the outbound SQL Server System, see the *SQL Server eWay Intelligent Adapter User's Guide*.

Configure the HTTP Servlet

You have the option of using the default Servlet URL for the HTTP server or using a different one.

For complete information on the servlet URL, see the *HTTP(S) eWay Intelligent Adapter User's Guide*.

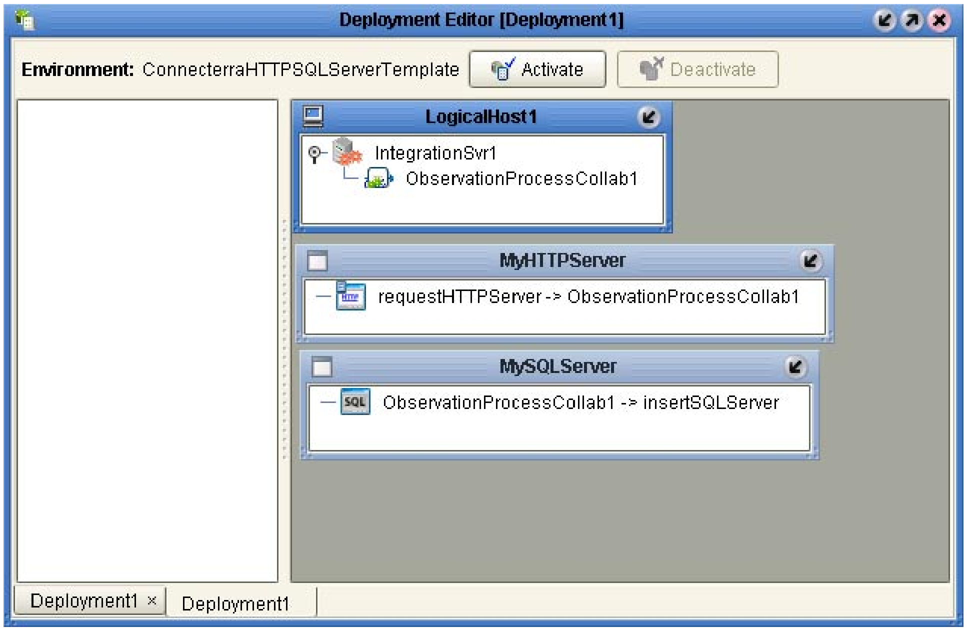
Create the Deployment Profile

1. In the Enterprise Explorer pane, click the Project Explorer tab.
2. Right-click the **ConnecterraHTTPSQLServerTemplate** Project, point to **New**, and then click **Deployment Profile**.
3. Type a **Deployment Profile Name**, select the appropriate **Environment**, and then click **OK**.

Make sure to select the Environment that you created in steps 2 and 3 of the previous procedure.

4 In the Deployment Profile, drag the components to the appropriate location as shown in the following figure.

Figure 65 Completed Deployment Profile

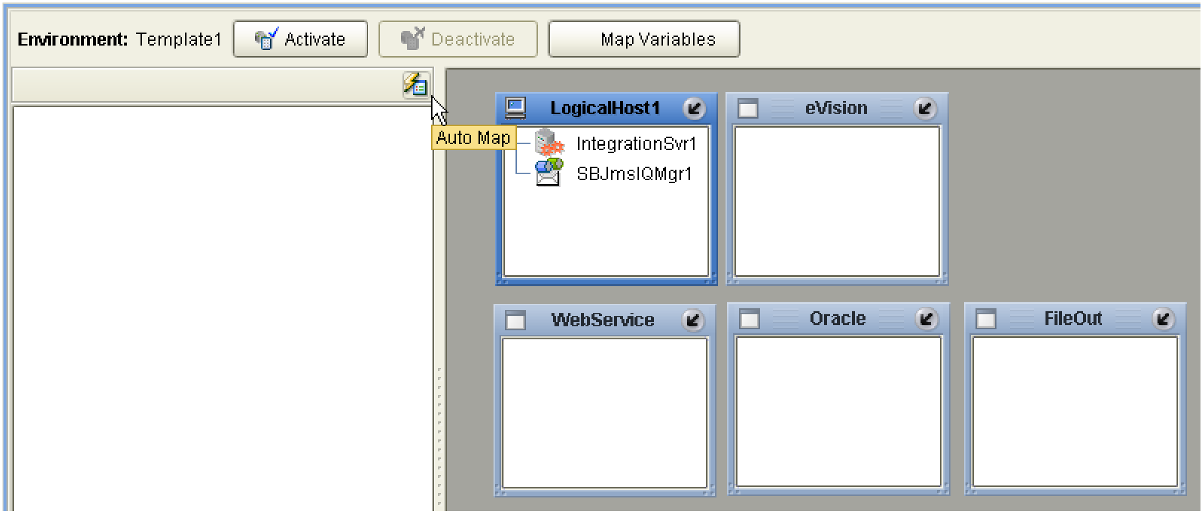


Environment: ConnecterraHTTPSQLServerTemplate

Deployment Editor [Deployment1]
Activate
Deactivate
LogicalHost1
IntegrationSvr1
ObservationProcessCollab1
MyHTTPServer
requestHTTPServer -> ObservationProcessCollab1
MySQLServer
ObservationProcessCollab1 -> insertSQLServer
Deployment1 ×
Deployment1

You may also use the **Auto Map** feature in **Enteprise Designer**’s **Environment Template**, by clicking the **AutoMap** button, to complete your Deployment Profile.

Figure 66 Auto Map Option



Environment: Template1

5 Click **Activate** to start the activation process.

6 Since your Logical Host is not yet running, click **No** to complete the activation without applying the changes to the Logical Host.

3.3.9 Using the ConnecterraHTTPTigerLogicTemplate Project

This Project uses ConnecTerra, HTTP input, and TigerLogic output. To use this Project template you will complete the following general steps:

- Run the New RFID Application Wizard.
- Create an Environment.
- Configure the HTTP Servlet.
- Configure the TigerLogic template.
- Create and activate a Deployment Profile.

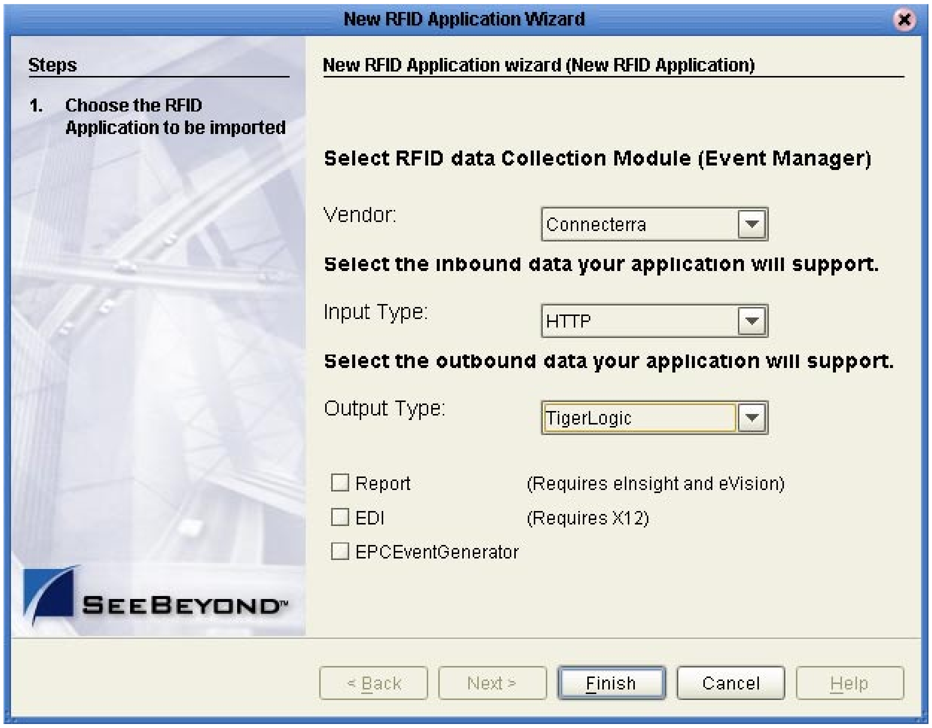
Create the ConnecterraHTTPTigerLogicTemplate Project template

1 Create a new Project in your Repository. This Project will act as the container Project for the RFID application.

2 Right-click the container Project, point to **New**, and then click **New RFID Application**.

3 Select the options shown in the following figure, and then click **Finish**.

Figure 67 ConnecterraHTTPTigerLogicTemplate Options



New RFID Application Wizard

New RFID Application wizard (New RFID Application)

Select RFID data Collection Module (Event Manager)

Vendor: Connecterra

Select the inbound data your application will support.

Input Type: HTTP

Select the outbound data your application will support.

Output Type: TigerLogic

□ Report (Requires eInsight and eVision)
□ EDI (Requires X12)
□ EPCEventGenerator

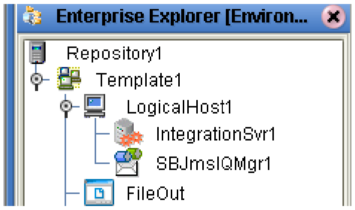
The New RFID Application Wizard creates a new Project template called **ConnecterraHTTPTigerLogicTemplate** in your container Project.

Create and configure the Environment

1. In the Enterprise Explorer pane, click the Environment Explorer tab.
2. Right-click your Repository, and then click **New Environment**.
3. Rename this Environment so it can be easily identified when you create the Deployment Profile (for example, **ConnecterraHTTPTigerLogicTemplate**).
4. Add the following items to your Environment:
   - Logical Host
   - SeeBeyond Integration Server
   - HTTP Server External System
   - File External System (outbound)

   Your Environment looks like the following figure.

Figure 68 ConnecterraHTTPTigerLogicTemplate Environment



5. Right-click **FileOut**, and then click **Properties** to view its associated values.

   The file external for your TigerLogic system only acts as a place holder.

Configure the HTTP Servlet

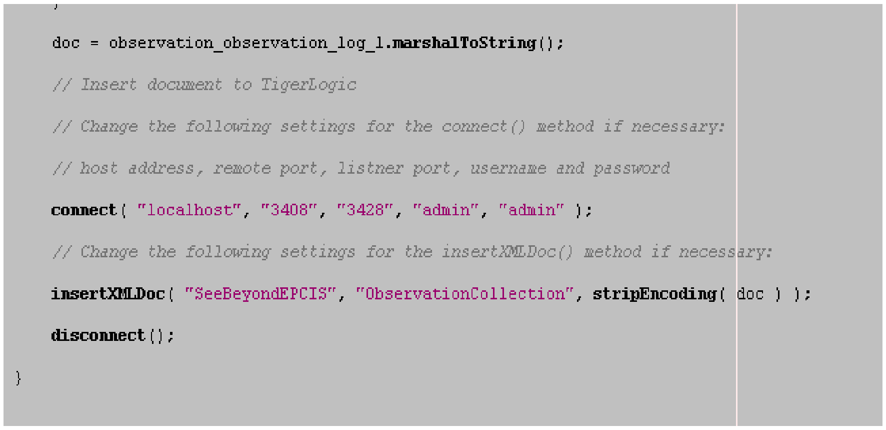
You have the option of using the default Servlet URL for the HTTP server or using a different one.

For complete information on the servlet URL, see the *HTTP(S) eWay Intelligent Adapter User's Guide*.

Configure the TigerLogic Template

1. In the Enterprise Explorer pane, click the Project Explorer tab.
2. Under the **ConnecterraHTTPTigerLogicTemplate** section, double-click **ObservationProcessCollab**.
3. To check out the collaboration, right-click **ObservationProcessCollab** and select **Check Out**. The Java source code appears in the right pane.
4. Navigate the source code until you find the `connect` call.

Figure 69 ObservationProcessCollab Check Out



```
    ;
    doc = observation_observation_log_1.marshalToString();

    // Insert document to TigerLogic

    // Change the following settings for the connect() method if necessary:

    // host address, remote port, listner port, username and password

    connect( "localhost", "3408", "3428", "admin", "admin" );

    // Change the following settings for the insertXMLDoc() method if necessary:

    insertXMLDoc( "SeeBeyondEPCIS", "ObservationCollection", stripEncoding( doc ) );

    disconnect();

}
```

5. Modify the parameter values used in the `connect` call if necessary. The call appears as

   ```
   connect( host address, remote port, listner port, username, password
   )
   ```

   where

   - `host address` is the TCP/IP address of the TigerLogic XML Server
   - `remote port` is the port number used by the TigerLogic Server
   - `listner port` is the port number of the client listener
   - `username` is the login username
   - `password` is the login password

   Thus, the `connect` call may appear as

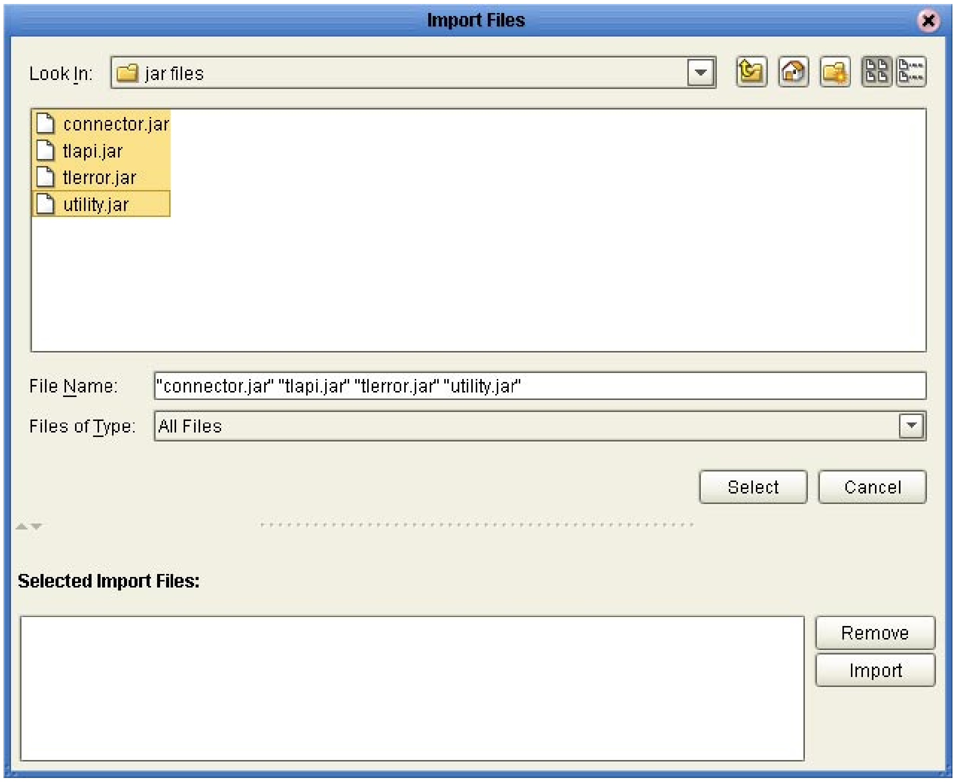
   ```
   connect( "localhost", "3408", "3428", "admin", "admin" )
   ```

6. Commit your changes by clicking the **Save** button on the **Enterprise Designer** toolbar.

7. To add your TigerLogic library to the **ObservationProcessCollab**, right-click the **ConnecterraHTTPTigerLogicTemplate** section, point to **New**, and then click **File**.

   The **Import Files** window appears.

Figure 70 Importing TigerLogic Library Files



Look In: jar files

connector.jar
tlapi.jar
tlerror.jar
utility.jar

File Name: "connector.jar" "tlapi.jar" "tlerror.jar" "utility.jar"

Files of Type: All Files

Selected Import Files:

8 Locate the following TigerLogic jar files:

- `connector.jar`
- `tlapi.jar`
- `utility.jar`

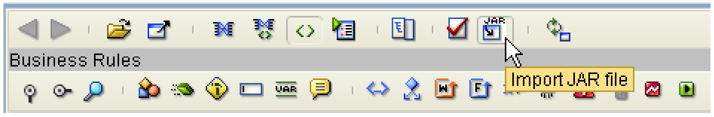
9 Highlight all three files simultaneously by holding the **Ctrl** key and clicking on each file name, click **Select,** and click **Import**.

The three jar files appear in the left pane under the **ConnecterraHTTPTigerLogicTemplate** section.

10 Double-click **ObservationProcessCollab** once again.

11 Click the **Import JAR file** button from the toolbar.

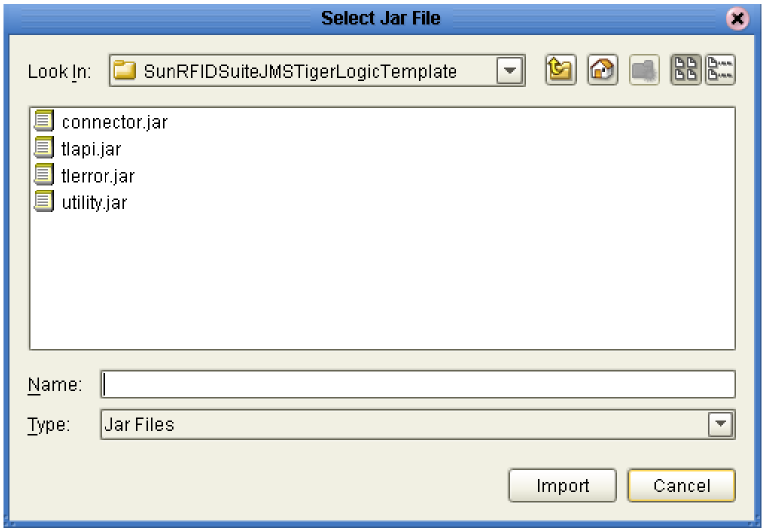
Figure 71 Import JAR File Button



12 In the **Add/Remove Jar Files** window, click **Add.**

13 In the **Select Jar File** window, locate the above-mentioned jar files in the folder of your **ConnecterraHTTPTigerLogicTemplate**.

Figure 72 Selecting Jar Files



SunRFIDSuiteJMSTigerLogicTemplate

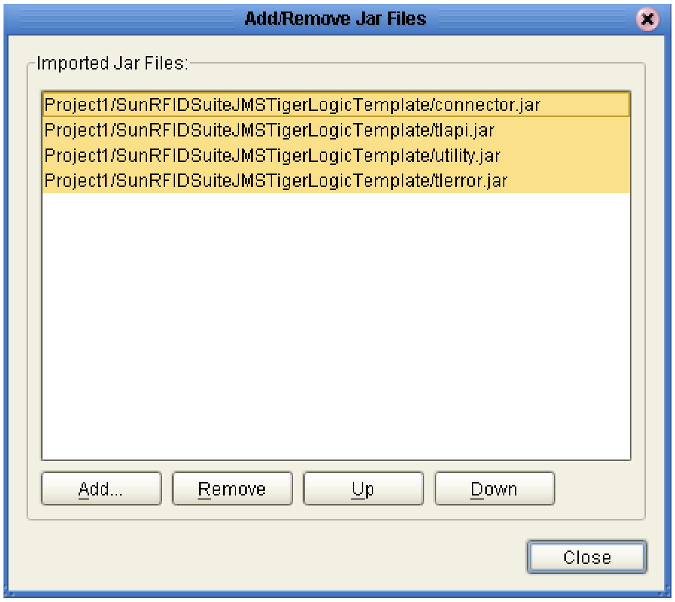
connector.jar
tlapi.jar
tlerror.jar
utility.jar

Jar Files

14 Select each filename and click **Import**.

The **Add/Remove Jar Files** window should be populated with all three file names.

Figure 73 Selecting Jar Files



Project1/SunRFIDSuiteJMSTigerLogicTemplate/connector.jar
Project1/SunRFIDSuiteJMSTigerLogicTemplate/tlapi.jar
Project1/SunRFIDSuiteJMSTigerLogicTemplate/utility.jar
Project1/SunRFIDSuiteJMSTigerLogicTemplate/tlerror.jar

15 Select all three files, as in the figure above, and click **Add**.

16 Commit your changes by clicking the **Save** button on the **Enterprise Designer** toolbar.

Create the Deployment Profile

1 In the Enterprise Explorer pane, click the Project Explorer tab.

2 Right-click the **ConnecterraHTTPTigerLogicTemplate** Project, point to **New**, and then click **Deployment Profile**.

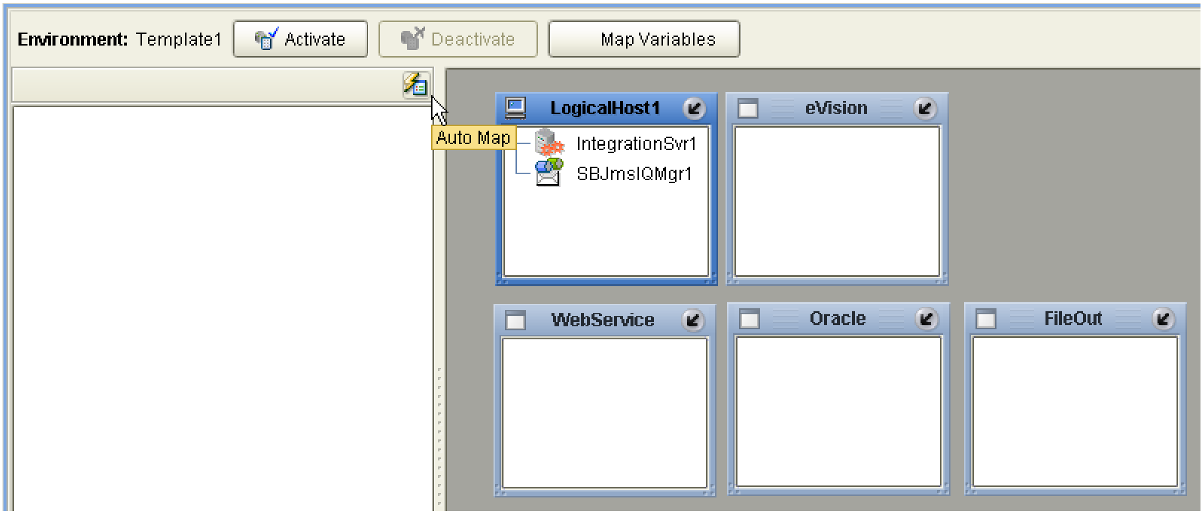
3 Type a **Deployment Profile Name**, select the appropriate **Environment**, and then click **OK**.

Make sure to select the Environment that you created in steps 2 and 3 of the previous procedure.

4 In the Deployment Profile, drag the components to the appropriate location, similar to the procedures for Oracle, DB2, and SQL Server Deployment Profiles.

You may also use the **Auto Map** feature in **Enteprise Designer**'s **Environment Template**, by clicking the **AutoMap** button, to complete your Deployment Profile.

Figure 74 Auto Map Option



5 Click **Activate** to start the activation process.

6 Since your Logical Host is not yet running, click **No** to complete the activation without applying the changes to the Logical Host.

Chapter 4

Creating the Observation_Log Database

What's in This Chapter

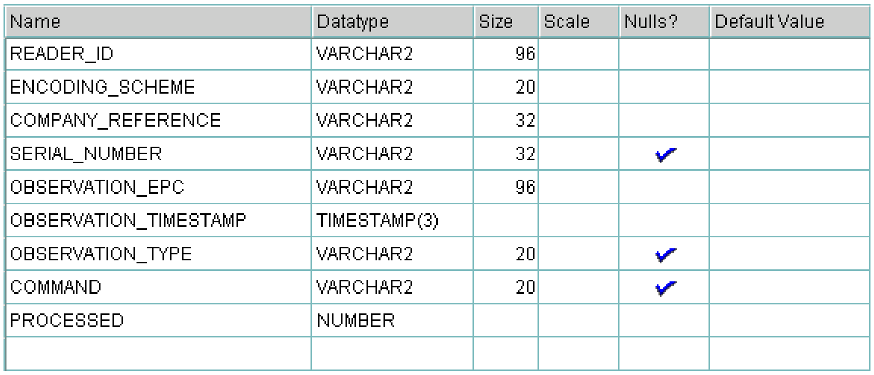
- "The Observation_Log Database" on page 84
- "Creating the Tables" on page 85
- "Archiving the Transactions" on page 86
- "Additional Tables for the Sample EPC IS Application" on page 87
- "Populating the Tables with Sample Data" on page 88

4.1 The Observation_Log Database

The RFID eWay uses a set of tables in a database to capture your RFID data. The eWay includes scripts for creating and archiving your data.

The figure below shows the columns and definitions for the table. Although the example shown here is for an Oracle database, the SQL Server or DB2 table appear similar.

Figure 75 Table Definition



Name	Datatype	Size	Scale	Nulls?	Default Value
READER_ID	VARCHAR2	96			
ENCODING_SCHEME	VARCHAR2	20			
COMPANY_REFERENCE	VARCHAR2	32			
SERIAL_NUMBER	VARCHAR2	32		✓	
OBSERVATION_EPC	VARCHAR2	96			
OBSERVATION_TIMESTAMP	TIMESTAMP(3)				
OBSERVATION_TYPE	VARCHAR2	20		✓	
COMMAND	VARCHAR2	20		✓	
PROCESSED	NUMBER				

4.1.1 Observation_Log SQL Files

You can create and archive the Observation_Log database using the two SQL scripts that are provided for you after you install the Enterprise Designer:

- **rfidcreate.sql**
- **rfidarc.sql**

These files can be found in one of the following locations, depending on which database you are using:

Table 6 SQL File Location

Database	Path
Oracle	<enterprise_designer_root>\edesigner\usrdir\modules\ext\rfidadapter\templates \Oracle
SQL Server	<enterprise_designer_root>\edesigner\usrdir\modules\ext\rfidadapter\templates \SQLServer
DB2	<enterprise_designer_root>\edesigner\usrdir\modules\ext\rfidadapter\templates \DB2

Note: *This location will vary slightly, depending on where you installed the Enterprise Designer on your local system.*

4.2 Creating the Tables

Use the script **rfidcreate.sql** to create the tables. See Table 6 for the location of this script.

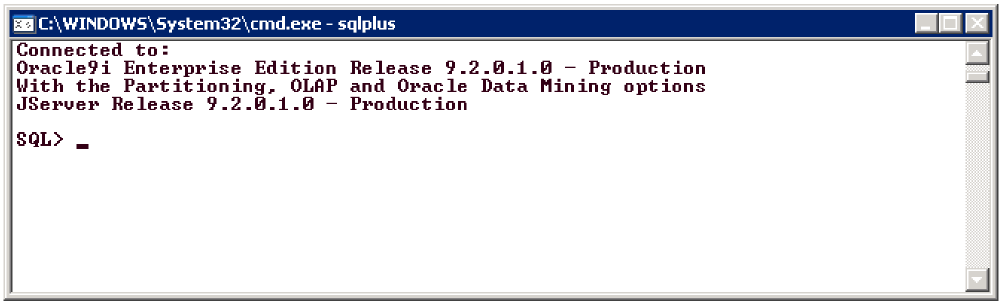
Note: *The following steps use the SQLPLUS command line utility. Your steps may vary if you use a different SQL utility.*

To create the tables

1. From a command prompt, type **SQLPLUS** and then press **Enter**.
2. Follow the login prompts to enter your user-name and password.

   The SQL prompt appears.

Figure 76 SQL Prompt



```
Connected to:
Oracle9i Enterprise Edition Release 9.2.0.1.0 - Production
With the Partitioning, OLAP and Oracle Data Mining options
JServer Release 9.2.0.1.0 - Production

SQL> _
```

3. To run the script, type **@rfidcreate.sql** and then press **Enter**.

The script attempts to drop the tables (in case they previously existed), creates the tables, and then exits SQLPlus when it is finished.

4.3 Archiving the Transactions

Use the script **rfidarc.sql** to archive the existing messages. See Table 6 for the location of this script.

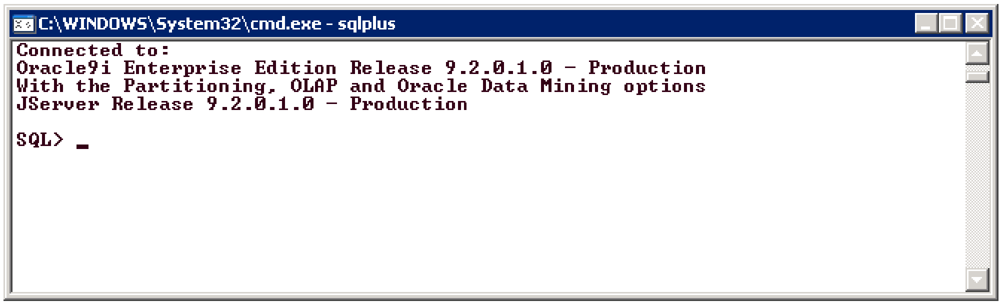
Note: *The following steps use the SQLPLUS command line utility. Your steps may vary if you use a different SQL utility.*

To archive the transactions

1. From a command prompt, type **SQLPLUS** and then press **Enter**.
2. Follow the login prompts to enter your user-name and password.

   The SQL prompt appears.

Figure 77 SQL Prompt



```
Connected to:
Oracle9i Enterprise Edition Release 9.2.0.1.0 - Production
With the Partitioning, OLAP and Oracle Data Mining options
JServer Release 9.2.0.1.0 - Production

SQL> _
```

3. To run the script, type **@rfidarc.sql** and then press **Enter**.

   You are then prompted to archive all messages or specific messages.
4. Press **1** to archive all messages or **2** to archive all processed messages after a specific timestamp.
   - If you choose to archive all messages, the script runs and exits SQLPlus when it is done.
   - If you choose to archive specific messages, enter the timestamp as you are prompted. The format that you enter for the timestamp depends on the timestamp format in your database. For example, Oracle uses the timestamp shown in the prompt, DB2 uses a format of `YYYY-MM-DD-HH-MI-SS.SSSSSS`, and the default setting for SQL Server is `MMM DD YYYY HH:MMAM/PM`. After you enter the timestamp, the script runs and exists SQLPlus.

   The archived records are written to the **OBSERVATION_ARCHIVE_LOG** table.

Note: *In some cases, you may actually need to modify the* ***rfidarc.sql*** *script to accommodate the date format for your database configuration.*

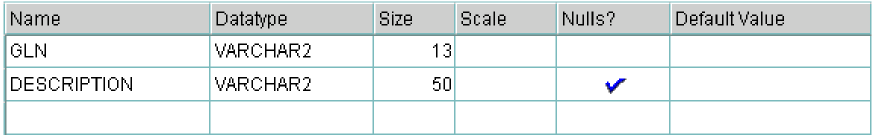
4.4 Additional Tables for the Sample EPC IS Application

In addition to the Observation_Log table, five additional tables are created for tracking and trace support. These tables are:

- VENDOR
- LOCATION
- PRODUCT
- READER
- ROUTING

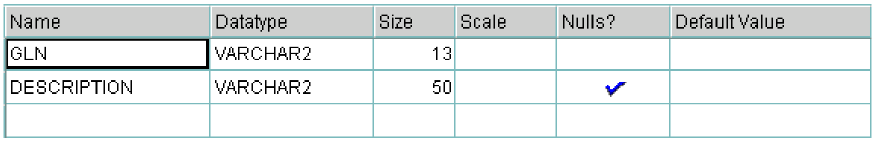
The following figures show these tables' columns and definitions:

Figure 78 VENDOR Definition



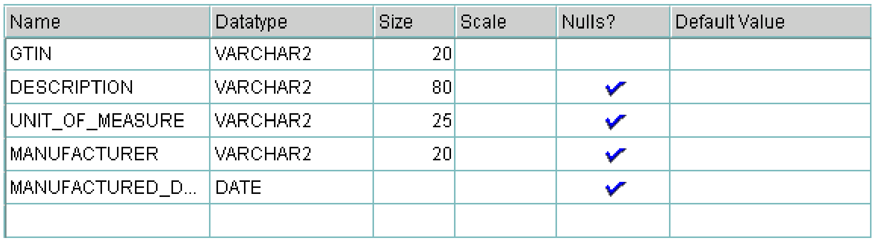
Name	Datatype	Size	Scale	Nulls?	Default Value
GLN	VARCHAR2	13			
DESCRIPTION	VARCHAR2	50		✓	

Figure 79 LOCATION Definition



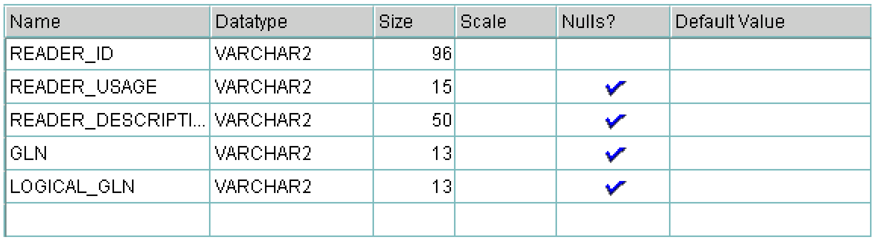
Name	Datatype	Size	Scale	Nulls?	Default Value
GLN	VARCHAR2	13			
DESCRIPTION	VARCHAR2	50		✓	

Figure 80 PRODUCT Definition



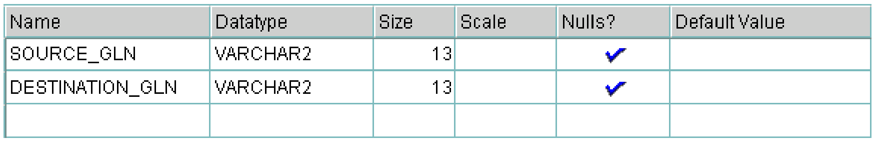
Name	Datatype	Size	Scale	Nulls?	Default Value
GTIN	VARCHAR2	20			
DESCRIPTION	VARCHAR2	80		✓	
UNIT_OF_MEASURE	VARCHAR2	25		✓	
MANUFACTURER	VARCHAR2	20		✓	
MANUFACTURED_D...	DATE			✓	

Figure 81 READER Definition



Name	Datatype	Size	Scale	Nulls?	Default Value
READER_ID	VARCHAR2	96			
READER_USAGE	VARCHAR2	15		✓	
READER_DESCRIPTI...	VARCHAR2	50		✓	
GLN	VARCHAR2	13		✓	
LOGICAL_GLN	VARCHAR2	13		✓	

Figure 82 ROUTING Definition



Name	Datatype	Size	Scale	Nulls?	Default Value
SOURCE_GLN	VARCHAR2	13		✓	
DESTINATION_GLN	VARCHAR2	13		✓	

4.5 Populating the Tables with Sample Data

The RFID eWay includes SQL scripts for you to use to populate your tables with sample data—or *seed* data. These scripts are located in **Oracle_Seed_Data.zip** file in **<*C:\ican50*>\edesigner\usrdir\modules\ext\rfidadapter\samples** directory.

Use SQLPlus to run these scripts and populate your tables with the seed data. Using the seed data makes it possible to view the sample reports without processing any actual RFID tag records.

Chapter 5

Working with the EPC Event Generator

What's in This Chapter

- **"What is the EPC Event Generator?" on page 89**
- **"Create a New RFID Application" on page 90**
- **"Configure EPCGeneratorOracleTemplate" on page 92**
- **"Configure EPCEventGenerator" on page 93**
- **"Run the Logical Host" on page 94**
- **"Run the Event Generator" on page 95**
- **"View the Results" on page 97**

5.1 What is the EPC Event Generator?

The EPC Event Generator is a sample application that simplifies the process of creating sample RFID tag records in a lab environment without the inconvenience of manually scanning RFID tags. The event generator is provided for testing and demonstration purposes.

The event generator reads records from an XML input file and publishes them to a queue. The flow of sample RFID records can be started, paused, resumed, and stopped through controls on a web page (when **Reports** is selected in the New RFID Application Wizard), or by command line (see **"Start the event generator from the command prompt" on page 96**). The output is written to the OBSERVATION_LOG database and simultaneously to a log file for easy viewing.

Note: *See* **Figure 83 on page 90** *for the exact options to use in the New RFID Application Wizard when using the EPC event generator.*

The process of installing, configuring, and running the event generator consists of these high-level steps:

1. Create the RFID application and the Environment
2. Configure the EPCGeneratorOracleTemplate Project
3. Configure the EPCEventGenerator Project
4. Run the Logical Host

5 Run the event generator

6 View the output results

These steps are covered in detail in the following sections in this chapter.

5.2 Create a New RFID Application

The EPC Event Generator is installed with your RFID application Project when you select the **Report** option in the **New RFID Application Wizard**. Before you select this option, make sure you have all of the required prerequisite ICAN products installed (see **“Prerequisite ICAN Modules” on page 13**).

To create the new RFID application

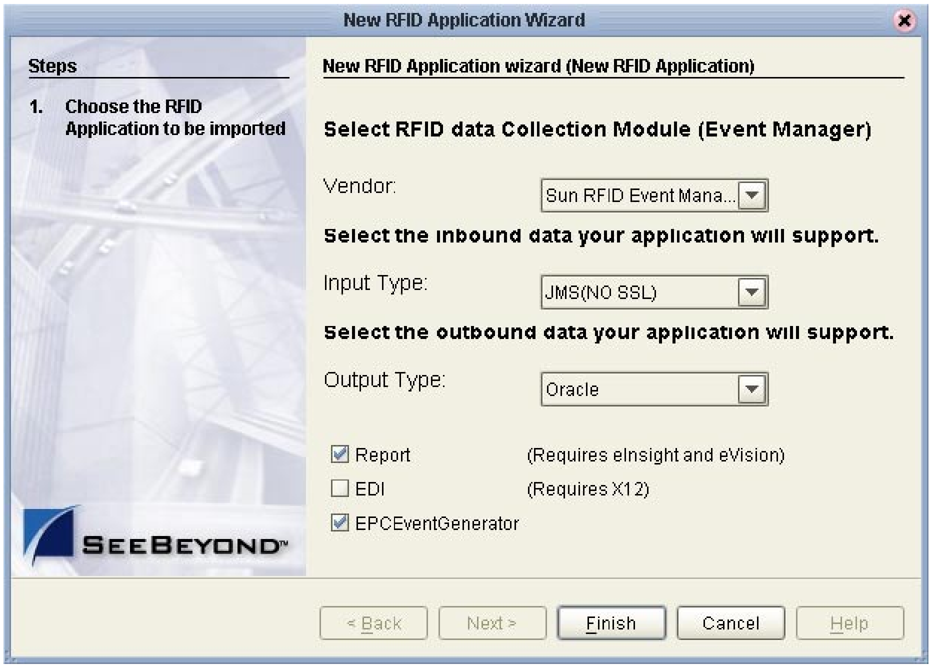
1 Create a new Project in your Repository. This Project will act as the container Project for this RFID application.

2 Right-click the container Project, point to **New**, and then click **New RFID Application**.

The New RFID Application Wizard appears.

3 Select the options shown in the following figure, and then click **Finish**.

Figure 83 New RFID Application Wizard Options



New RFID Application Wizard

Steps

1. Choose the RFID Application to be imported

New RFID Application wizard (New RFID Application)

Select RFID data Collection Module (Event Manager)

Vendor: Sun RFID Event Mana...

Select the inbound data your application will support.

Input Type: JMS(NO SSL)

Select the outbound data your application will support.

Output Type: Oracle

☑ Report (Requires eInsight and eVision)

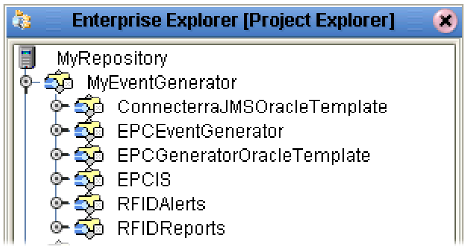
☐ EDI (Requires X12)

☑ EPCEventGenerator

In the New RFID Application Wizard, you *must* select the options shown in the previous figure.

After you click **Finish**, the wizard will take slightly longer to finish than it does without the **Reports** option selected. With this option selected, the wizard creates a collection of Projects in your container Project. The results are shown in the following figure.

Figure 84 EPC Event Generator Project



Enterprise Explorer [Project Explorer]

MyRepository
MyEventGenerator
ConnecterraJMSOracleTemplate
EPCEventGenerator
EPCGeneratorOracleTemplate
EPCIS
RFIDAlerts
RFIDReports

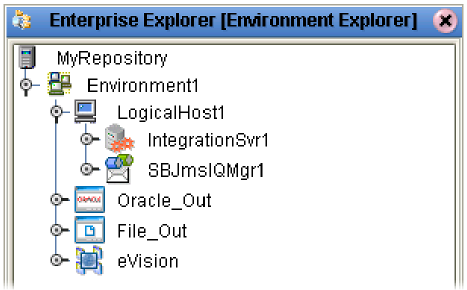
After the wizard is finished, the next step is to add an Environment for the event generator.

Create and configure the Environment

1. In the Enterprise Explorer pane, click the Environment Explorer tab.
2. Right-click your Repository, and then click **New Environment**.
3. Add the following items to your Environment:
   - Logical Host
   - SeeBeyond Integration Server
   - SeeBeyond JMS IQ Manager
   - Oracle External System (outbound)
   - File External System (outbound)
   - eVision External System

   Your Environment looks like the following figure.

Figure 85 Environment



Enterprise Explorer [Environment Explorer]

MyRepository
Environment1
LogicalHost1
IntegrationSvr1
SBJmsIQMgr1
Oracle_Out
File_Out
eVision

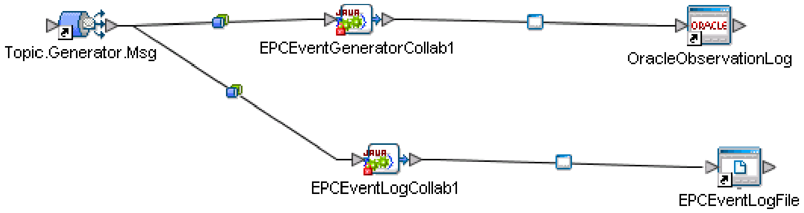
4. Right-click the Oracle external, and then click **Properties**.
5. Configure the Oracle external as appropriate for your Oracle system, and then click **OK** when finished.

   For complete details on configuring the outbound Oracle System, see the *Oracle eWay Intelligent Adapter User's Guide*.

5.3 Configure EPCGeneratorOracleTemplate

The EPCGeneratorOracleTemplate loads the RFID messages from an inbound topic, publishes them an external Oracle system, and logs the transactions in a file.

Figure 86 Connectivity Map



This Project requires little configuration on your part—configure the output path of the File eWay (optional) and create and activate a Deployment Profile.

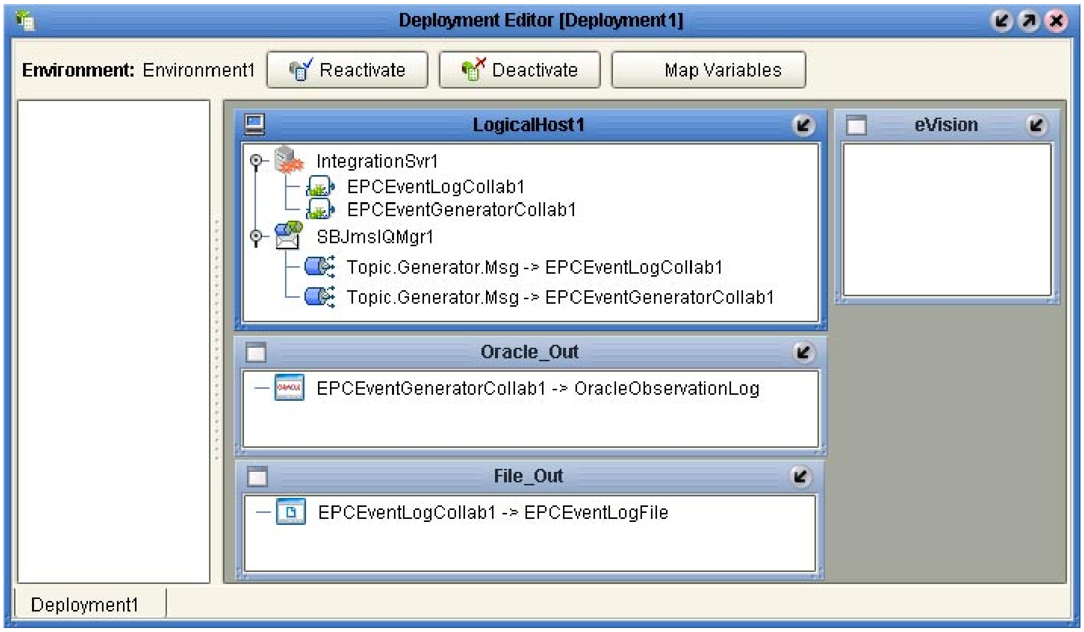
Configure the EPCGeneratorOracleTemplate

1. In the Enterprise Explorer pane, click the Project Explorer tab.
2. Double-click the **EPCGeneratorOracleTemplate** Project to expand it and view its contents.
3. The **EPCEventLogFile** File eWay is pre-configured to write the log file to **c:\temp**. You can change this if you need to publish the output to a different location.

   For complete instructions on configuring the file eWay, see the *File eWay Intelligent Adapter User's Guide*.

   To modify the file eWay, you must first check out the Connectivity Map from the Repository. Remember to check it back in after modifying the file eWay.
4. To create the Deployment Profile, right-click the **EPCGeneratorOracleTemplate** Project, point to **New**, and then click **Deployment Profile**.
5. Type the **Deployment Profile Name**, select the **Environment** that you created in the previous procedure, and then click **OK**.
6. In the Deployment Profile, drag the components to the appropriate location as shown in the following figure.

Figure 87 Deployment Profile


Deployment Editor [Deployment1]
Environment: Environment1
Reactivate
Deactivate
Map Variables
LogicalHost1
IntegrationSvr1
EPCEventLogCollab1
EPCEventGeneratorCollab1
SBJmsIQMgr1
Topic.Generator.Msg -> EPCEventLogCollab1
Topic.Generator.Msg -> EPCEventGeneratorCollab1
eVision
Oracle_Out
EPCEventGeneratorCollab1 -> OracleObservationLog
File_Out
EPCEventLogCollab1 -> EPCEventLogFile
Deployment1

7 Click **Activate** to start the Activation process.

8 Since your Logical Host is not yet running, click **No** to complete the activation without applying the changes to the Logical Host.

5.4 Configure EPCEventGenerator

The EPCEventGenerator Project is used to start, pause, and stop the flow of sample messages to the topic in the EPCGeneratorOracleTemplate. This Project uses an eInsight Business Process for logical flow control and an eVision web page as a front-end.

This Project requires little configuration on your part—you need to change the URL Business Process and create and activate a Deployment Profile.

Configure the EPCEventGenerator Template

1 In the Enterprise Explorer pane, click the Project Explorer tab.

2 Double-click the **EPCEventGenerator** Project to expand it and view its contents.

3 Right-click the **ControlFlow** Business Process and then click **Check Out**. Click **OK** to continue the Version Control Check Out process.

4 Right-click **ControlFlow** and then click **Properties**.

5 Edit the **Target Namespace** as appropriate for your system. For example, for a Repository named **MyRepository**, you would use the following URL:

```
http://localhost:12000/repository/MyRepository/EPCEventGenerator/
MyRepository/PageFlow1
```

6 Click **OK** to save the properties for the business process.

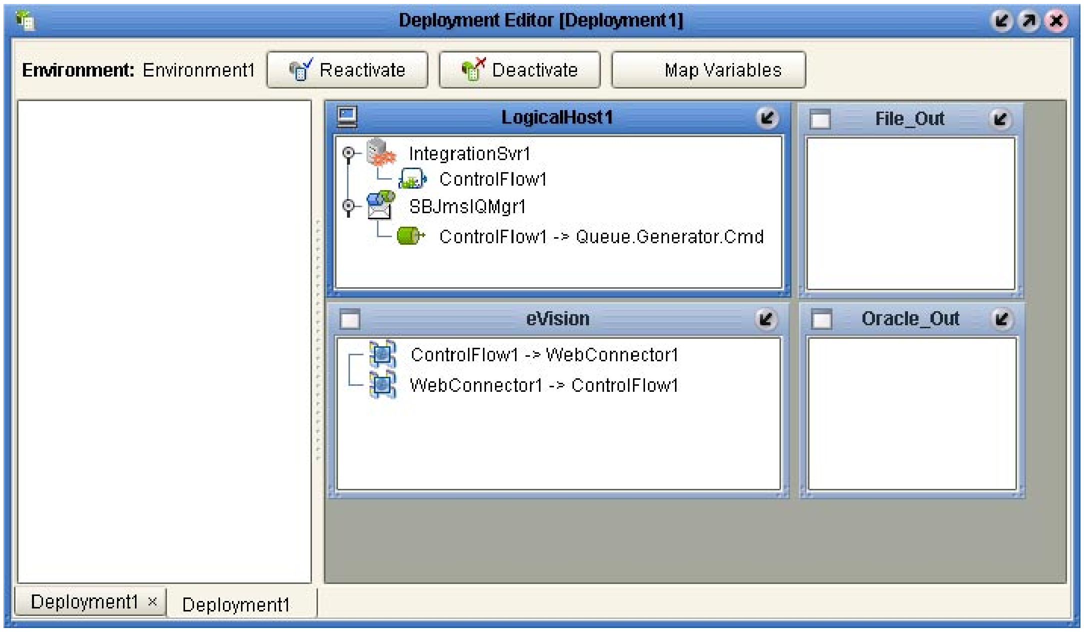
7 Right-click **ControlFlow** and then click **Check In** to check the updated properties into the Repository.

8 To create the Deployment Profile, right-click the **EPCEventGenerator** Project, point to **New**, and then click **Deployment Profile**.

9 Type the **Deployment Profile Name**, select the Environment that you created previously, and then click **OK**.

10 In the Deployment Profile, drag the components to the appropriate location as shown in the following figure.

Figure 88 Deployment Profile


Deployment Editor [Deployment1]
Environment: Environment1
Reactivate
Deactivate
Map Variables
LogicalHost1
IntegrationSvr1
ControlFlow1
SBJmsIQMgr1
ControlFlow1 -> Queue.Generator.Cmd
File_Out
eVision
ControlFlow1 -> WebConnector1
WebConnector1 -> ControlFlow1
Oracle_Out
Deployment1 ×
Deployment1

11 Click **Activate** to start the Activation process.

12 Since your Logical Host is not yet running, click **No** to complete the activation without applying the changes to the Logical Host.

13 Make a note of the URL that displays for your EPC Event Generator Control page.

5.5 Run the Logical Host

The next step is to start the Logical Host. In this example, the Logical Host will host both of the Deployment Profiles.

Start the Logical Host

1 If you have not previously done so, download and install the Logical Host from the Enterprise Manager.

2 From a Command Prompt, navigate to the **bootstrap\bin** directory for your Logical Host—for example, **C:\Ican50\logicalhost\bootstrap\bin**.

3 To start your Logical Host, type the following command:

```
bootstrap -r repository_url -e environment_name -l logical_host
-i user_id -p password
```

where

repository_url is the URL (including Repository name) for your Repository server. For example, **http://MyServer:15000/MyRepository**. Substitute the name and port for your Repository as well as the name of your Repository.

environment_name is the name of your Environment (for example, Environment1),

logical_host is the name of your Logical Host (for example, LogicalHost1), and

username/password are the user name and password for your Repository.

An example for this command is:

```
bootstrap -r localhost -e Environment1 -l LogicalHost1
-i Administrator -p STC
```

Since this is the first time running this Logical Host, the Logical Host will connect to the Repository and download all of the Project-specific files that it requires. While this runs in the background, you can continue to the next procedure.

5.6 Run the Event Generator

The event generator is a utility that reads an input file (**events.xml**) and publishes these messages to a JMS topic. These messages are then processed by the subscribers in the EPCGeneratorOracleTemplate Project. Before you can start the event generator, you must first copy the event generator files to a working location, verify that you have the prerequisite .jar files in your CLASSPATH, configure the event generator properties, and optionally edit the input data file.

Copy the event generator files

1 Using Windows Explorer, navigate to the location where the event generator files are temporarily stored. This location depends on where you installed your Enterprise Designer.

If you installed the Enterprise Designer in **C:\Ican50**, then the files will be in **C:\Ican50\edesigner\usrdir\modules\ext\rfidadapter\generator**.

2 Copy these files to a working directory, such as **C:\MyEventGenerator**.

Verify the prerequisite .jar files

- Verify that you have all of the files listed in the SeeBeyond EPC Event Generator section of Table 4 in **“JMS Requirements” on page 13**. Either copy these files to the same location you copied the event generator files to or copy them to some other location referenced by your CLASSPATH environment variable.

Configure the properties file

1. Navigate to the location where you copied the event generator files—for example, **C:\MyEventGenerator**.
2. Use a text editor, such as Windows Notepad, to open **event.properties**.
3. Locate the line that begins with **jms.server.name** and add the name of the machine where you will run the event generator.
4. Locate the line that begins with **jms.server.port** and add the port number for JMS server.
5. Optionally edit the remaining properties as appropriate.
6. Save the **event.properties** file and close the text editor.

Optionally edit input file

- The input file, events.xml, includes a number of sample records. If you wish, you can change the contents of this input file or substitute it with one of your own.

Start the event generator via its web page

1. Before starting the event generator, verify that the Logical Host started successfully and is ready. You can check the status by viewing the console window where you ran the bootstrap command or by using the Monitor in the Enterprise Manager.
2. Navigate to the directory where you copied the event generator files and run the **rungenerator.bat** batch file.

Note: *If the event generator fails to start, verify that you have all of the required .jar files. Also, the batch file may not run if your you do not have the correct version of* ***Java.exe****. If this is the case, modify the batch file to use* ***Java.exe*** *located in your* ***<enterprise_designer_root>\edesigner\jdk\jre\bin*** *directory.*

3. In a web browser window, navigate to the URL that displayed when you activated the EPCEventGenerator Deployment Profile. Use the controls in the web page to start, pause, resume, or stop the event generator.

Start the event generator from the command prompt

You may also start the EPCEventGenerator at the command prompt by navigating to the location where the event generator files are temporarily stored. This location depends on where you installed your Enterprise Designer. If you installed the Enterprise Designer in **C:\Ican50**, then the files will be in **C:\Ican50\edesigner\usrdir\modules\ext\rfidadapter\generator**. Copy the **runcmd.bat** file from that directory to the same location as the **rungenerator.bat** file and run it. You may issue the EPCEventGenerator commands from the command prompt by typing `runcmd <command>`, where the commands are `start`, `stop`, `pause`, or `resume`. If the command prompt is used, the eInsight Business Process and the eVision web page are not used.

5.7 View the Results

The results of the records processed by the event generator are written to your Oracle table (SYSTEM.OBSERVATION_LOG) and to a log file in the directory you specified in step 3 of the procedure in **"Configure EPCGeneratorOracleTemplate" on page 92**.

Use the Oracle Enterprise Manager Console to view the records in SYSTEM.OBSERVATION_Log. Additionally use a text editor to view the results in the output text file.

Chapter 6

Using the RFID eWay Reports

What's in This Chapter

- **"The RFID Reports for the Sample EPC IS Application" on page 98**
- **"Activating Reports for the Sample EPC IS Application" on page 99**
- **"Inventory Report" on page 106**
- **"OverStock Report" on page 108**
- **"Transit Time Report" on page 110**
- **"EPC Trace Report" on page 112**
- **"PO Reconciliation by Item Report" on page 115**
- **"PO Reconciliation by SSCC Report" on page 117**
- **"Using the Worklist Manager Alerts" on page 119**

6.1 The RFID Reports for the Sample EPC IS Application

The RFID eWay includes six reports. These reports are:

- Inventory
- Overstock
- Transit Time
- EPC Trace
- PO Reconciliation by Item
- PO Reconciliation by SSCC

To use these reports, you must select the Report option in the New RFID Application Wizard when you create your RFID application.

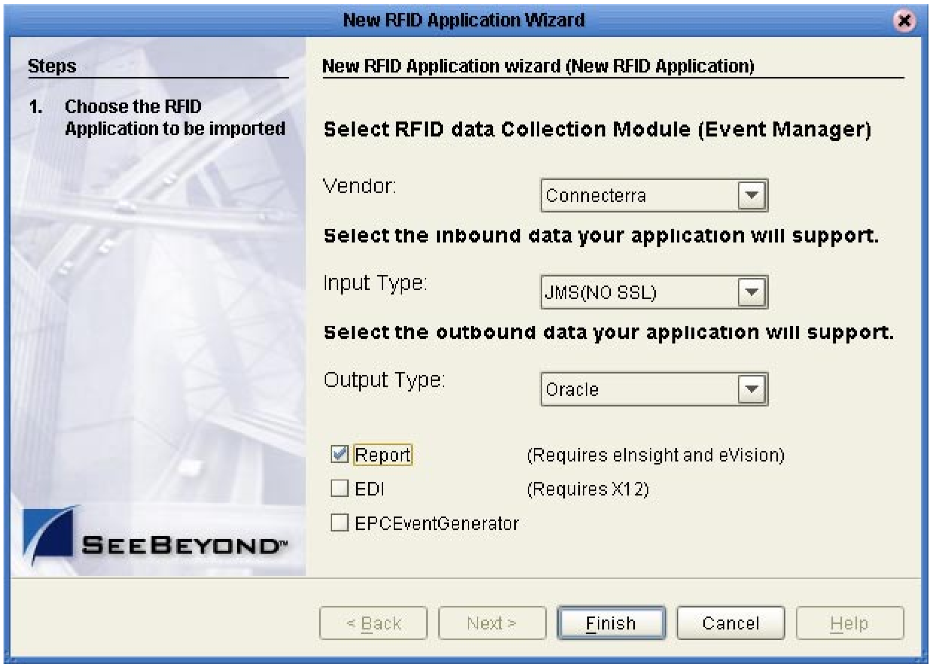
Viewing sample reports

To view these sample reports without processing any RFID tag records, you can load sample data into your database. See **"Populating the Tables with Sample Data" on page 88** for more information.

6.2 Activating Reports for the Sample EPC IS Application

When you create a new RFID application, choose whether to add the reports to the Project. When you select the Reports option (see the following figure), the wizard includes the **RFIDReports** Project in your RFID application.

Figure 89 Reports Option



New RFID Application Wizard

Steps

1. Choose the RFID Application to be imported

New RFID Application wizard (New RFID Application)

Select RFID data Collection Module (Event Manager)

Vendor: Connecterra

Select the inbound data your application will support.

Input Type: JMS(NO SSL)

Select the outbound data your application will support.

Output Type: Oracle

☑ Report (Requires eInsight and eVision)

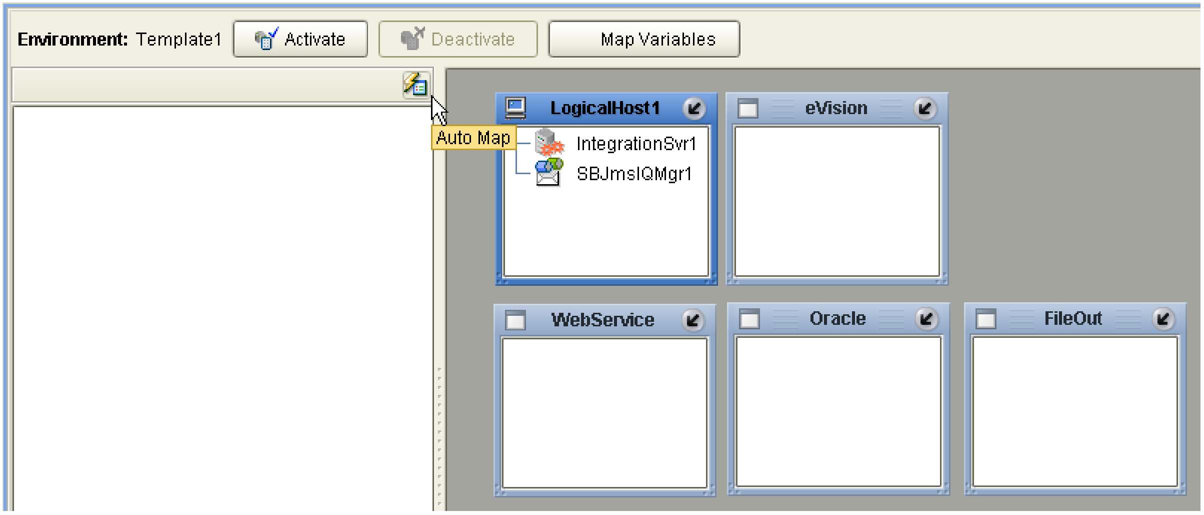
☐ EDI (Requires X12)

☐ EPCEventGenerator

Configuring the **RFIDReports** Project requires only a few extra steps beyond the configuration of your RFID Project. When you create the Environment, also add an eVision external system. When you add the Deployment Profile to the **RFIDReports** Project, drag each of the components to the appropriate container in the Deployment Profile. Drag the Services to the Integration Server, the Web Connectors to the eVision external, and so forth.

You may also use the **Auto Map** feature in **Enteprise Designer**’s **Environment Template**, by clicking the **AutoMap** button, to complete your Deployment Profile.

Figure 90 Auto Map Option



When you activate the Deployment Profile, make a note of the URLs for the reports server. These are the URLs which you will use to view the reports.

It is important to note that the eVision web pages are pre-configured to use the default port 18004 for the Integration Server and “localhost” for hostname. If you choose to change the port for the Integration Server or use another hostname, you will also need to reconfigure the following eVision pages:

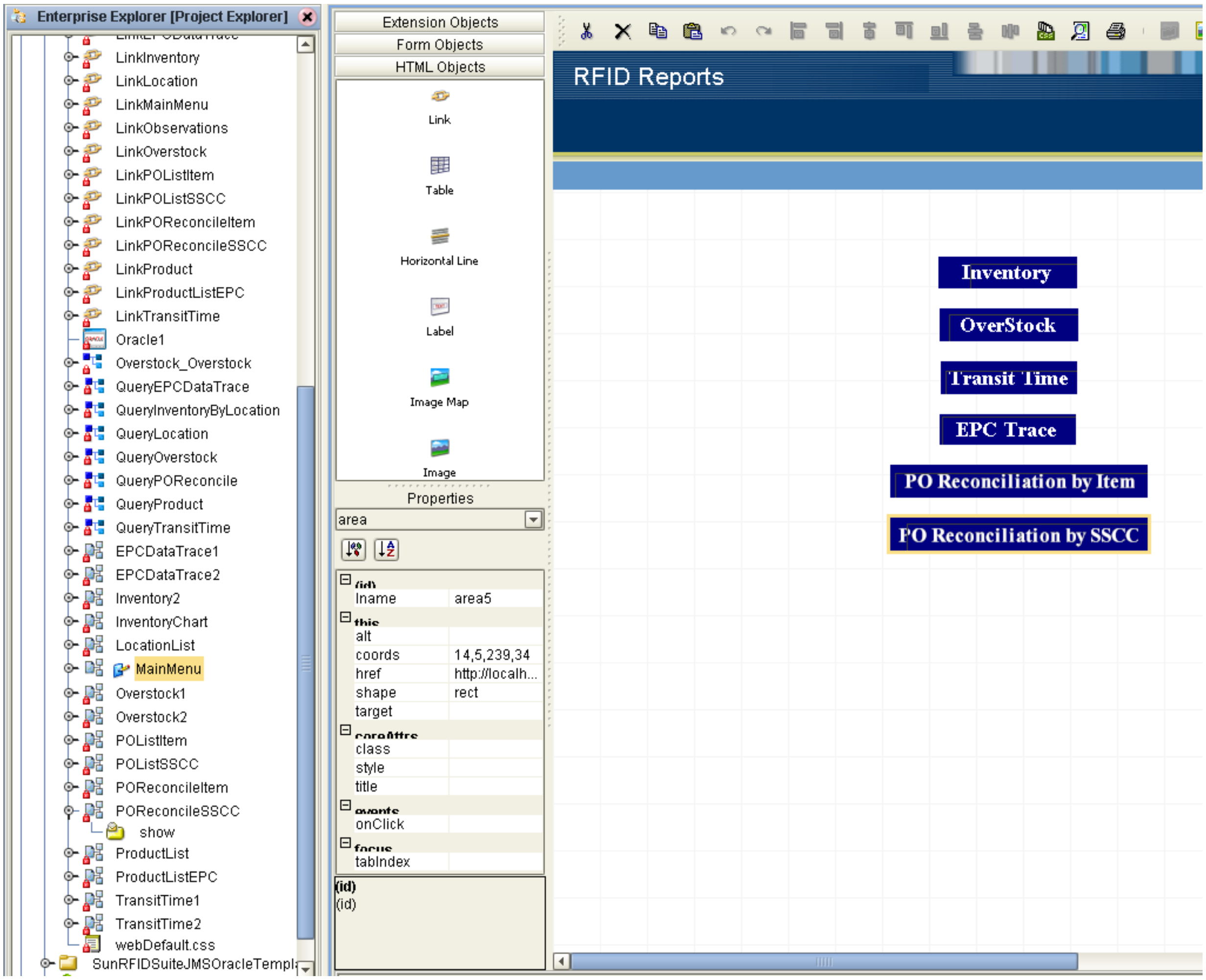
- **MainMenu**
- **LocationList**
- **Overstock1**
- **POListItem**
- **POListSSCC**
- **ProductList**
- **ProductListEPC**

Configuring eVision Main Menu page

1. In the Enterprise Explorer pane, click the Project Explorer tab.
2. Under the **RFIDReports** folder of your project, double-click the **MainMenu** section to open the **MainMenu** page.

   The page will appear on the right pane.

Figure 91 eVision Main Menu Page Configuration



(id)
Iname area5
this
alt
coords 14,5,239,34
href http://localh...
shape rect
target
coreAttrs
class
style
title
events
onClick
focus
tabIndex

(id)
(id)

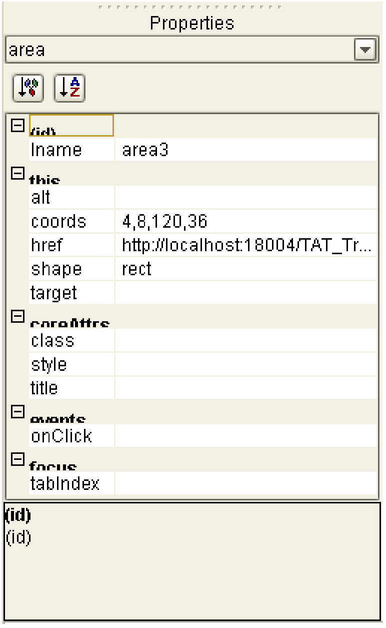
3 Right-click the **MainMenu** section and click **Check Out**.

4 Click **OK** at the **Version** Control Prompt.

5 Click the page button (such as **PO Reconciliation by SSCC**).

The properties for the page link appear in the **Properties** pane.

Figure 92 eVision Page Properties



Properties

area

(id)	
Iname	area3
this	
alt	
coords	4,8,120,36
href	http://localhost:18004/TAT_Tr...
shape	rect
target	
coreAttrs	
class	
style	
title	
events	
onClick	
focus	
tabIndex	

(id)
(id)

6 Edit the host name and port number values in the **href** line.

7 Save your changes by clicking the **Save** button on the **Enterprise Designer** toolbar.

8 Edit the host name and port number values for the rest of the buttons on the **Main Menu** page by repeating the steps above.

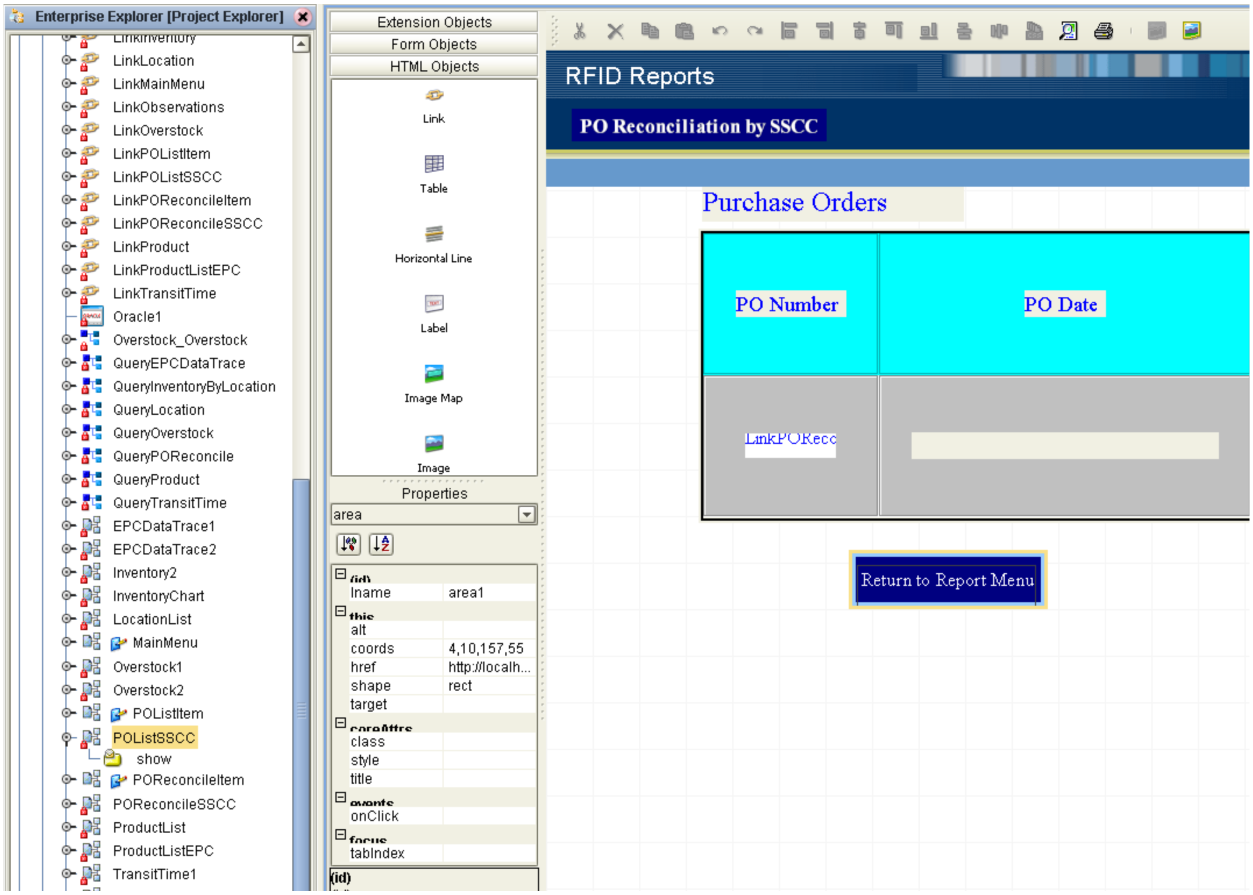
Configuring eVision Reports pages

Each of the reports includes a **Return to Report Menu** button. This button includes a link back to the main reports page, such as the **PO Reconciliation by SSCC** page shown in **Figure 108 on page 117**. If you chose to change the default port number for the main reports page, then the **Return to Report Menu** buttons in the reports will also need to be re-configured to use the new port number.

1 For each of the eVision pages you wish to reconfigure (such as **PO Reconciliation by SSCC**), double-click on its assigned section (such as **POReconcileSSCC**) under the **RFIDReports** folder of your project.

The page will appear on the right pane.

Figure 93 eVision PO Reconciliation by SSCC Page Configuration



area

(id)

lname	area1

this

alt	
coords	4,10,157,55
href	http://localh...
shape	rect
target	

coreAttrs

class	
style	
title	

events

onClick	

focus

tabIndex	

(id)

2 Right-click the **MainMenu** section and click **Check Out**.

3 Click **OK** at the **Version** Control Prompt.

4 Click the **Return to Report Menu** button.

The properties for the page link appear in the **Properties** pane.

5 Edit the host name and port number values in the **href** line.

6 Save your changes by clicking the **Save** button on the **Enterprise Designer** toolbar.

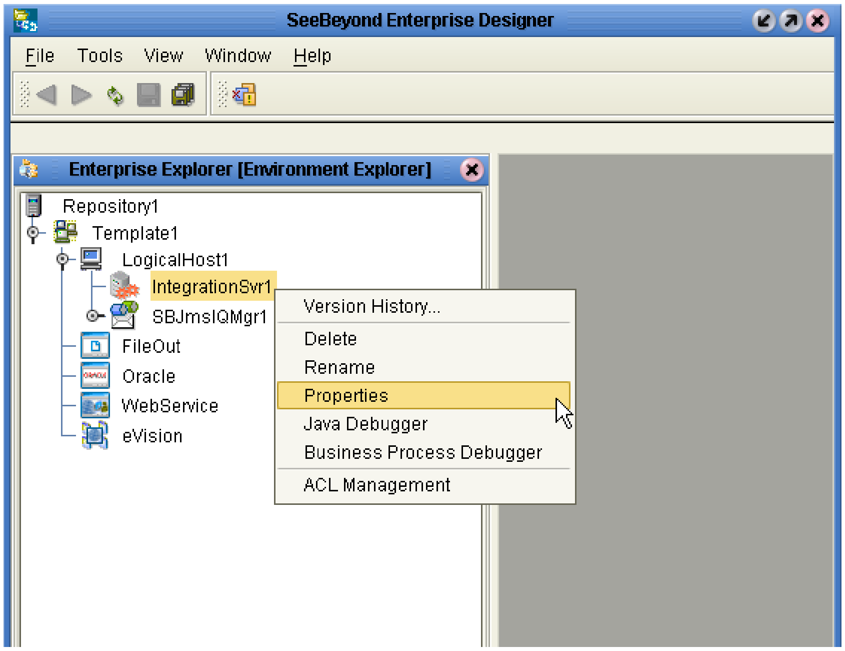
7 Edit the host name and port number values for the **Return to Report Menu** button in the rest of the eVision pages for RFID Reports by repeating the steps above.

Reviewing the port for the Integration Server

1 In the Enterprise Explorer pane, click the Environment Explorer tab.

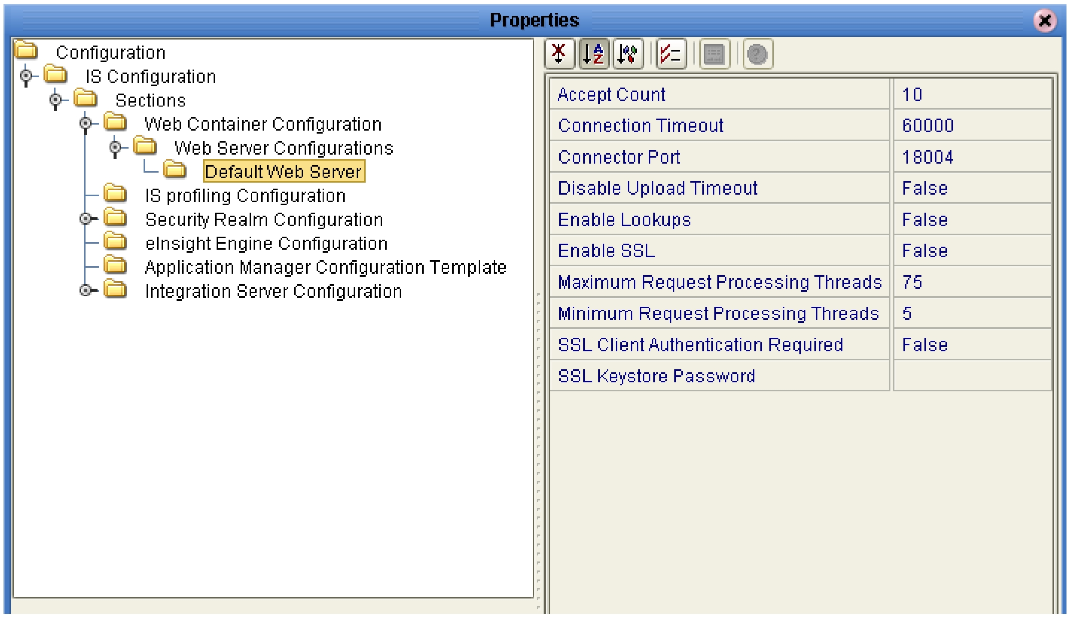
2 Right-click the **Integration Server**, and then click **Properties**.

Figure 94 Integration Server Properties



3 In the **Properties** window, click the section **Configuration** > **IS Configuration** > **Sections** > **Web Container Configuration** > **Web Server Configurations** > **Default Web Server**. The value of the port is stored in the **Connector Port** parameter.

Figure 95 Integration Server Parameters

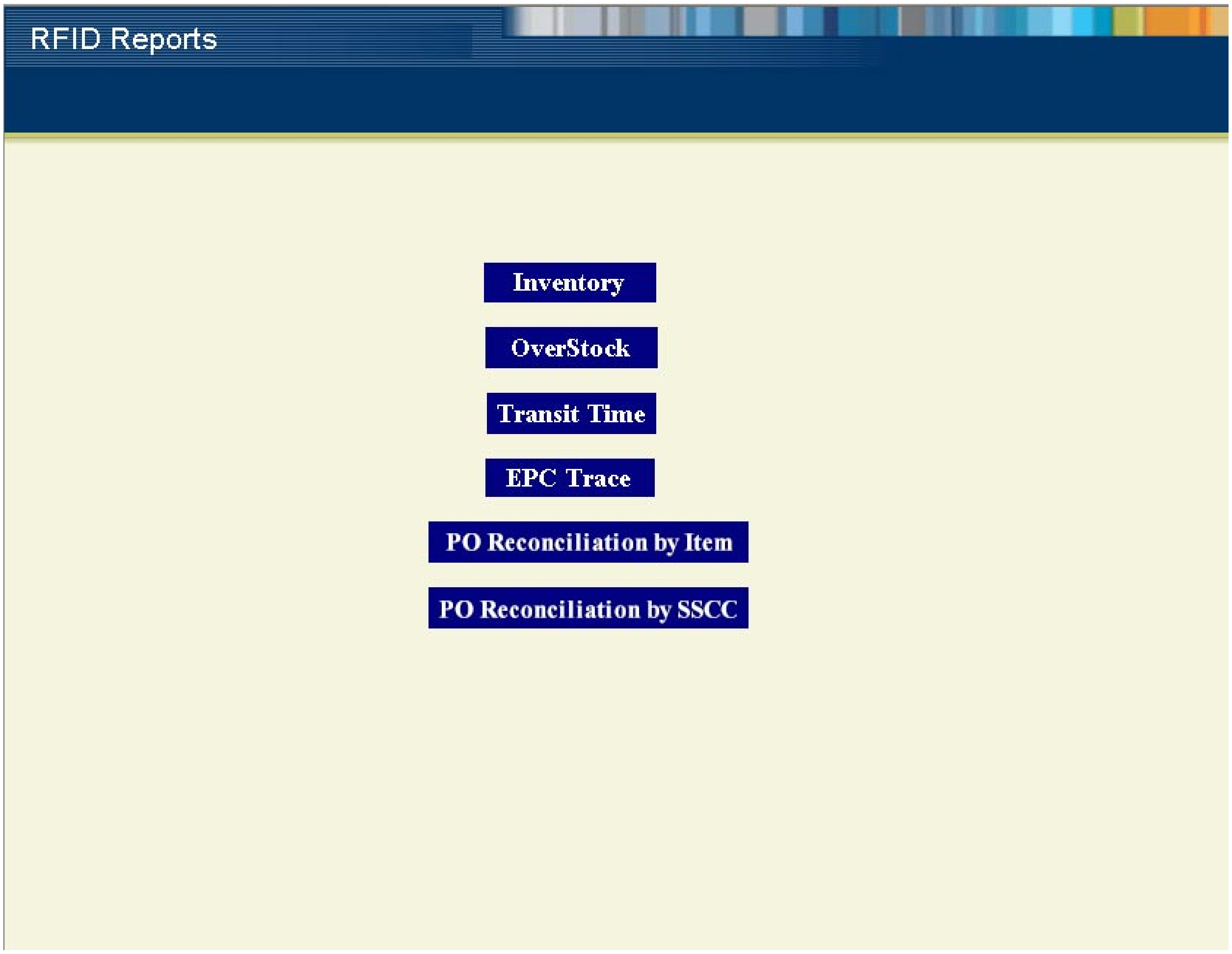

Properties
Configuration
IS Configuration
Sections
Web Container Configuration
Web Server Configurations
Default Web Server
IS profiling Configuration
Security Realm Configuration
eInsight Engine Configuration
Application Manager Configuration Template
Integration Server Configuration

Accept Count	10
Connection Timeout	60000
Connector Port	18004
Disable Upload Timeout	False
Enable Lookups	False
Enable SSL	False
Maximum Request Processing Threads	75
Minimum Request Processing Threads	5
SSL Client Authentication Required	False
SSL Keystore Password	

Viewing the reports

1. Using your web browser, navigate to the URL for your repository server (this is URL that was displayed during the Deployment Profile activation).
2. From the RFID Reports page, select the report you wish to view (see the following figure).

Figure 96 RFID Reports page



6.3 Inventory Report

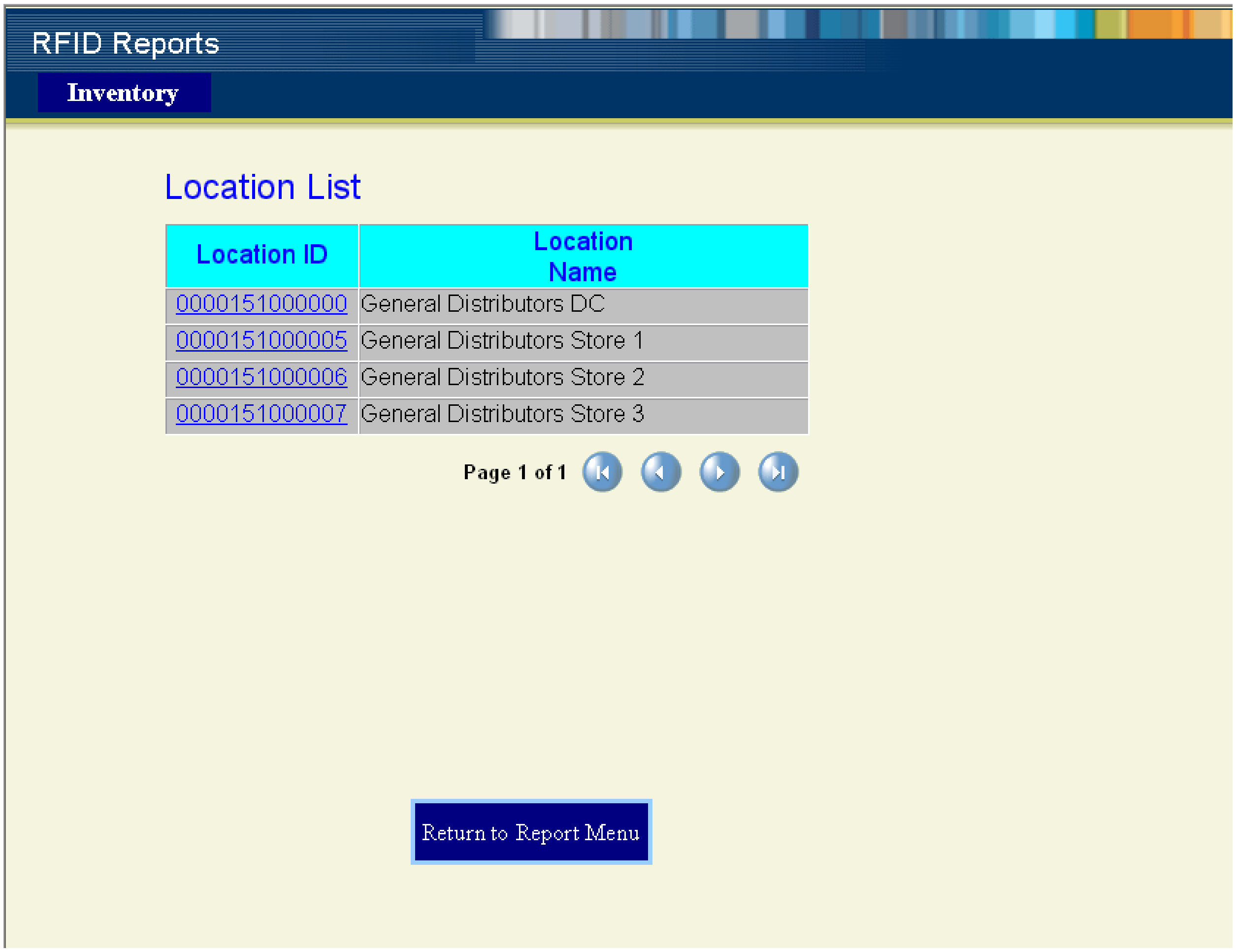
The Inventory report shows the quantity in stock for the products at the selected location.

To view the Inventory Report

1 From the RFID Reports page, click **Inventory**.

The location list appears.

Figure 97 Inventory Location List



Location List

Location ID	Location Name
0000151000000	General Distributors DC
0000151000005	General Distributors Store 1
0000151000006	General Distributors Store 2
0000151000007	General Distributors Store 3

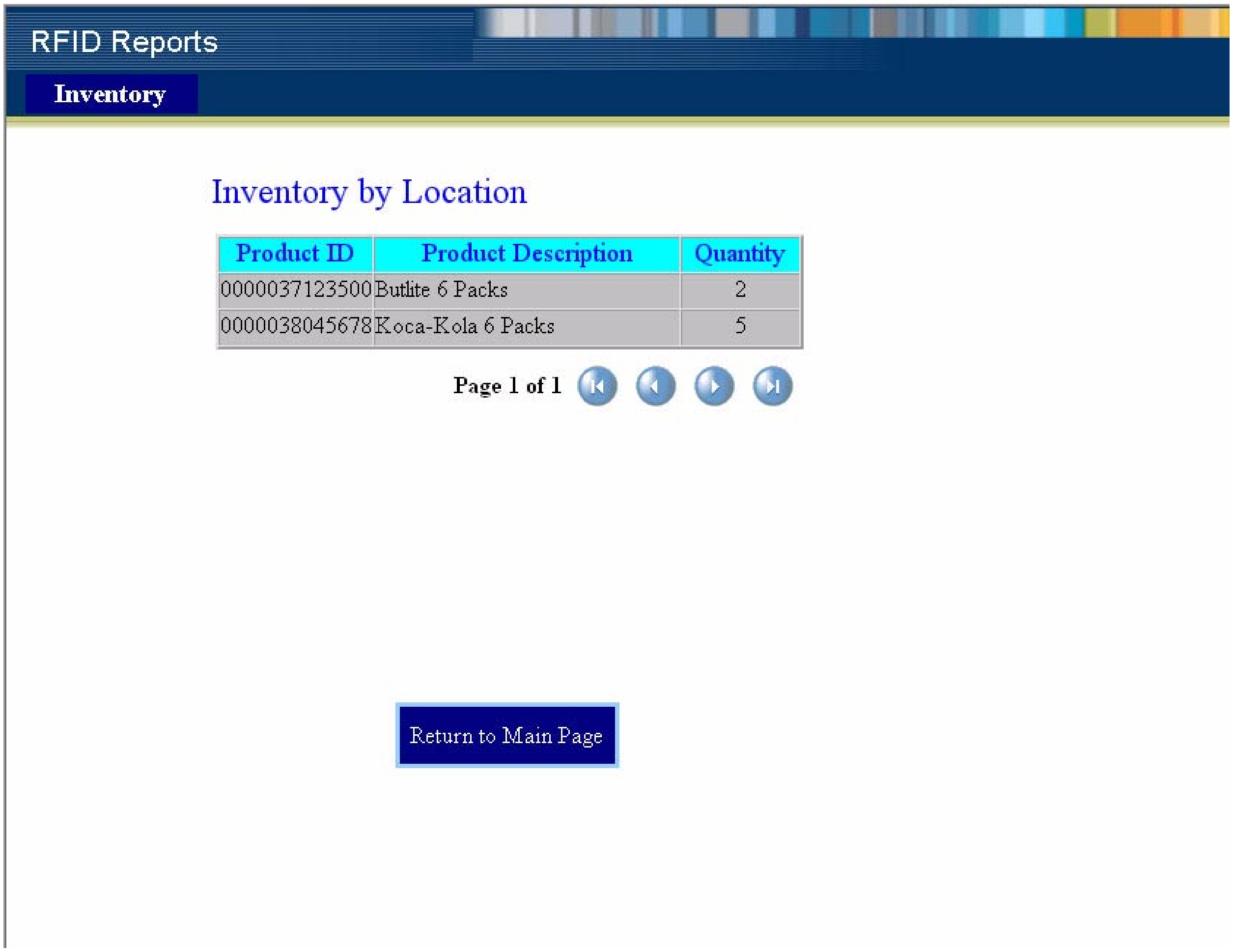
Page 1 of 1

If the location list exceeds one page, you may navigate it by clicking the buttons at the bottom of the list.

2 From the location list, select the location you wish to view by clicking the appropriate **Location ID** link.

The Inventory report appears.

Figure 98 Inventory Report



Inventory by Location

Product ID	Product Description	Quantity
0000037123500	Butlite 6 Packs	2
0000038045678	Koca-Kola 6 Packs	5

Page 1 of 1

If the Inventory report exceeds one page, you may navigate it by clicking the buttons at the bottom of the report.

6.4 OverStock Report

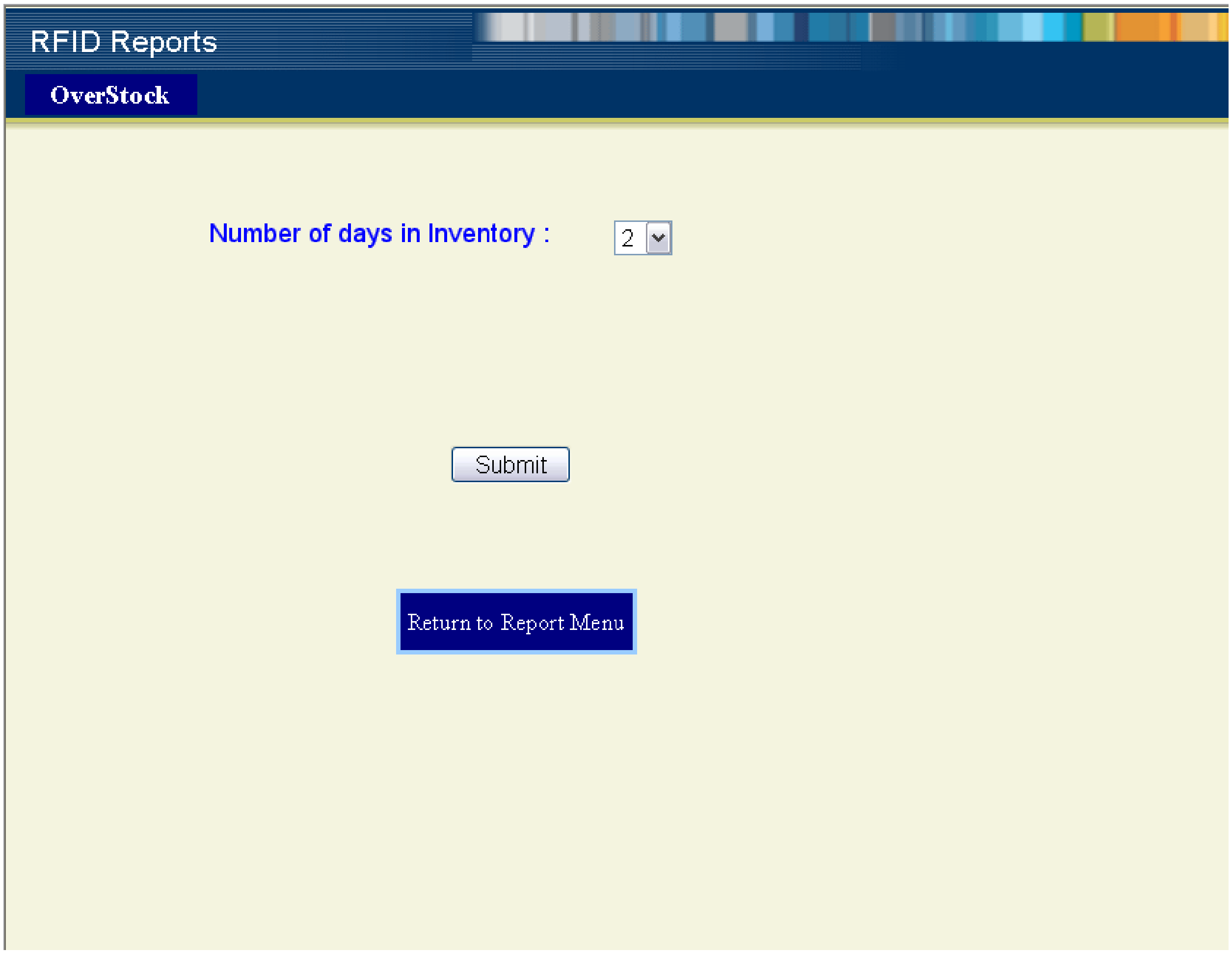
The OverStock report shows the products that have been on hand longer than the number of days you specify.

To view the OverStock Report

1 From the RFID Reports page, click **Overstock**.

The following page appears.

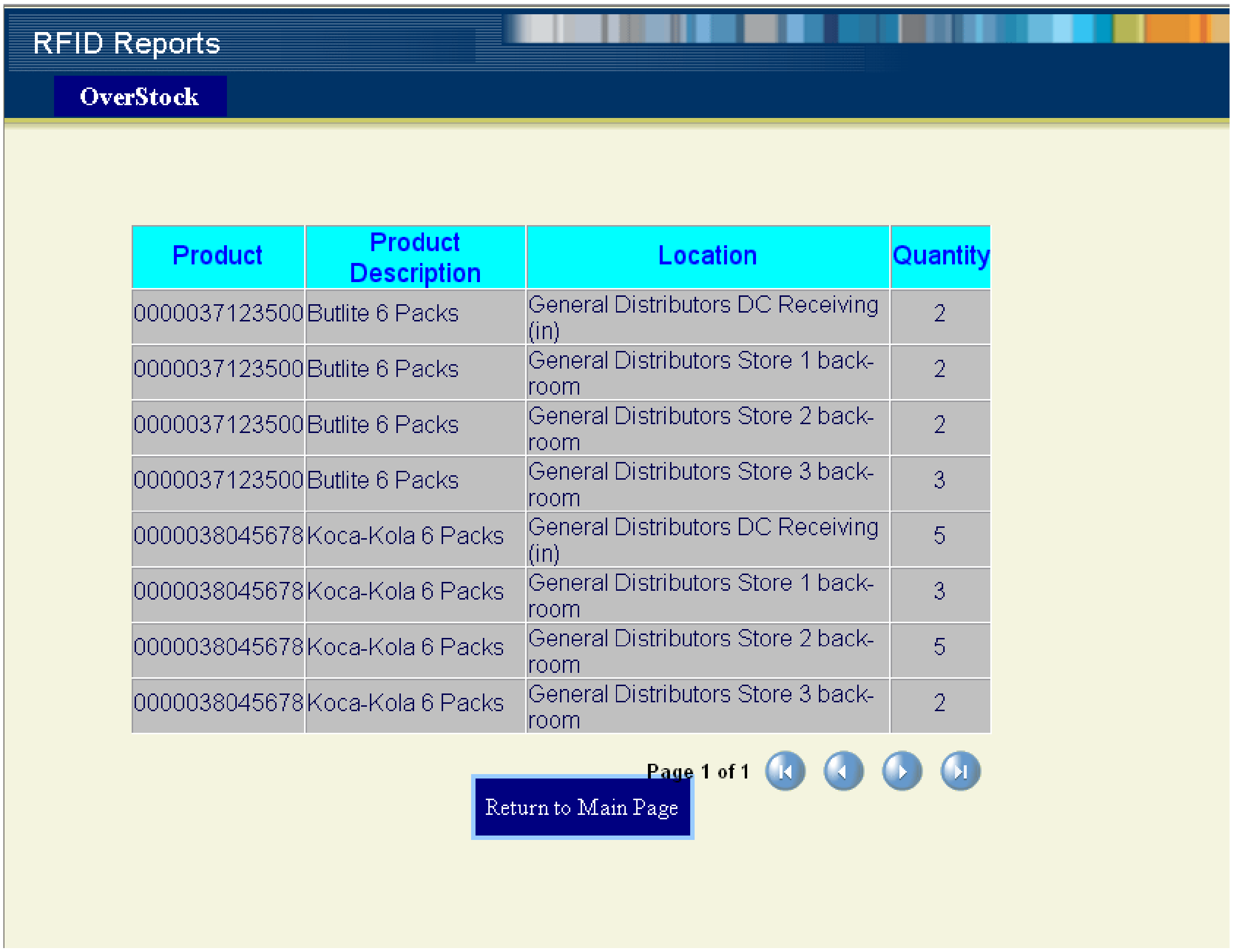
Figure 99 Overstock Number of Days in Inventory Page



2 Select the number of days in Inventory from the drop-down menu, and click **Submit.**

The OverStock report appears.

Figure 100 OverStock Report



Product	Product Description	Location	Quantity
0000037123500	Butlite 6 Packs	General Distributors DC Receiving (in)	2
0000037123500	Butlite 6 Packs	General Distributors Store 1 back-room	2
0000037123500	Butlite 6 Packs	General Distributors Store 2 back-room	2
0000037123500	Butlite 6 Packs	General Distributors Store 3 back-room	3
0000038045678	Koca-Kola 6 Packs	General Distributors DC Receiving (in)	5
0000038045678	Koca-Kola 6 Packs	General Distributors Store 1 back-room	3
0000038045678	Koca-Kola 6 Packs	General Distributors Store 2 back-room	5
0000038045678	Koca-Kola 6 Packs	General Distributors Store 3 back-room	2

If the Overstock report exceeds one page, you may navigate it by clicking the buttons at the bottom of the report.

6.5 Transit Time Report

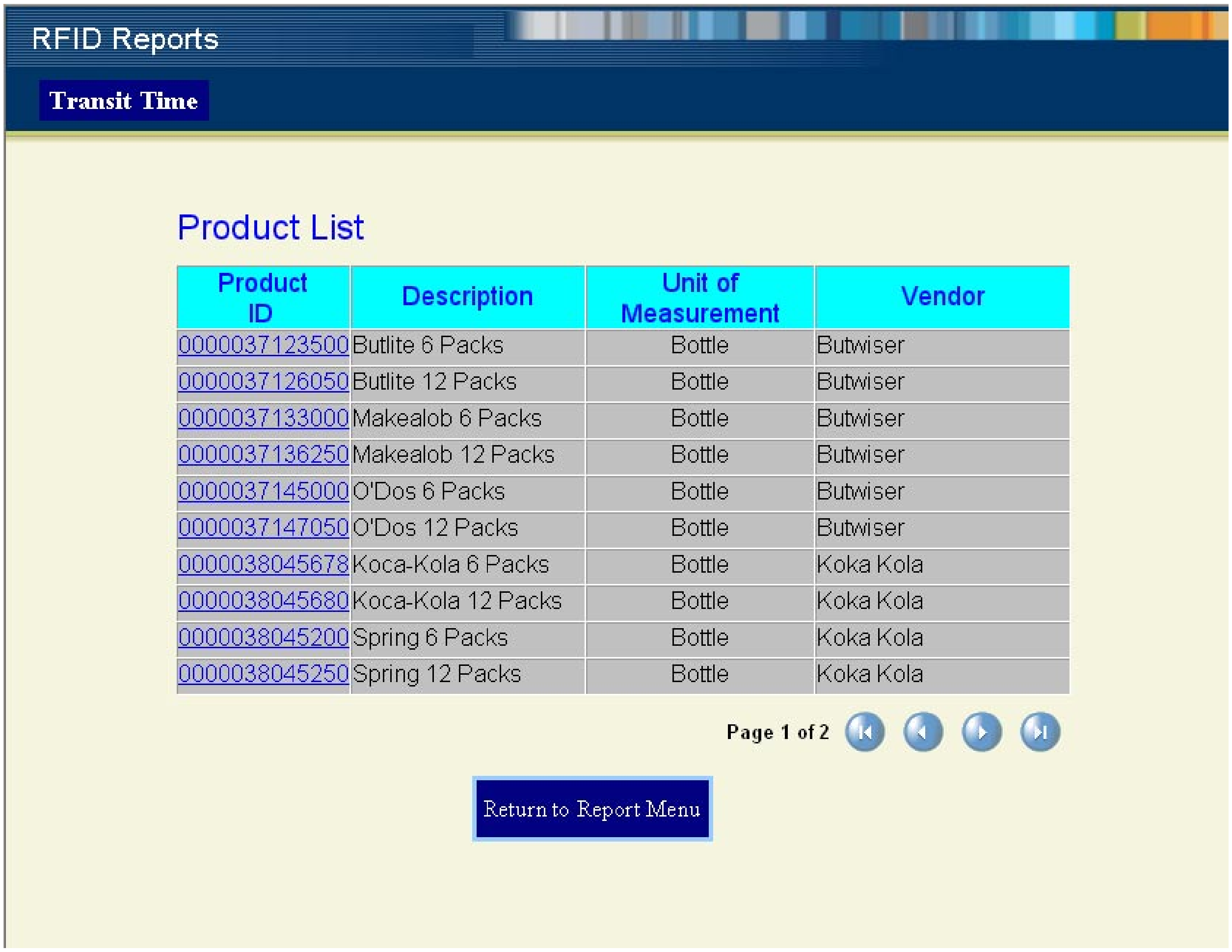
The Transit Time report shows the average time that it takes for a product to travel from one location to another.

To view the Transit Time Report

1. From the RFID Reports page, click **Transit Time**.

   The product list appears.

Figure 101 Transit Time Product List



Product List

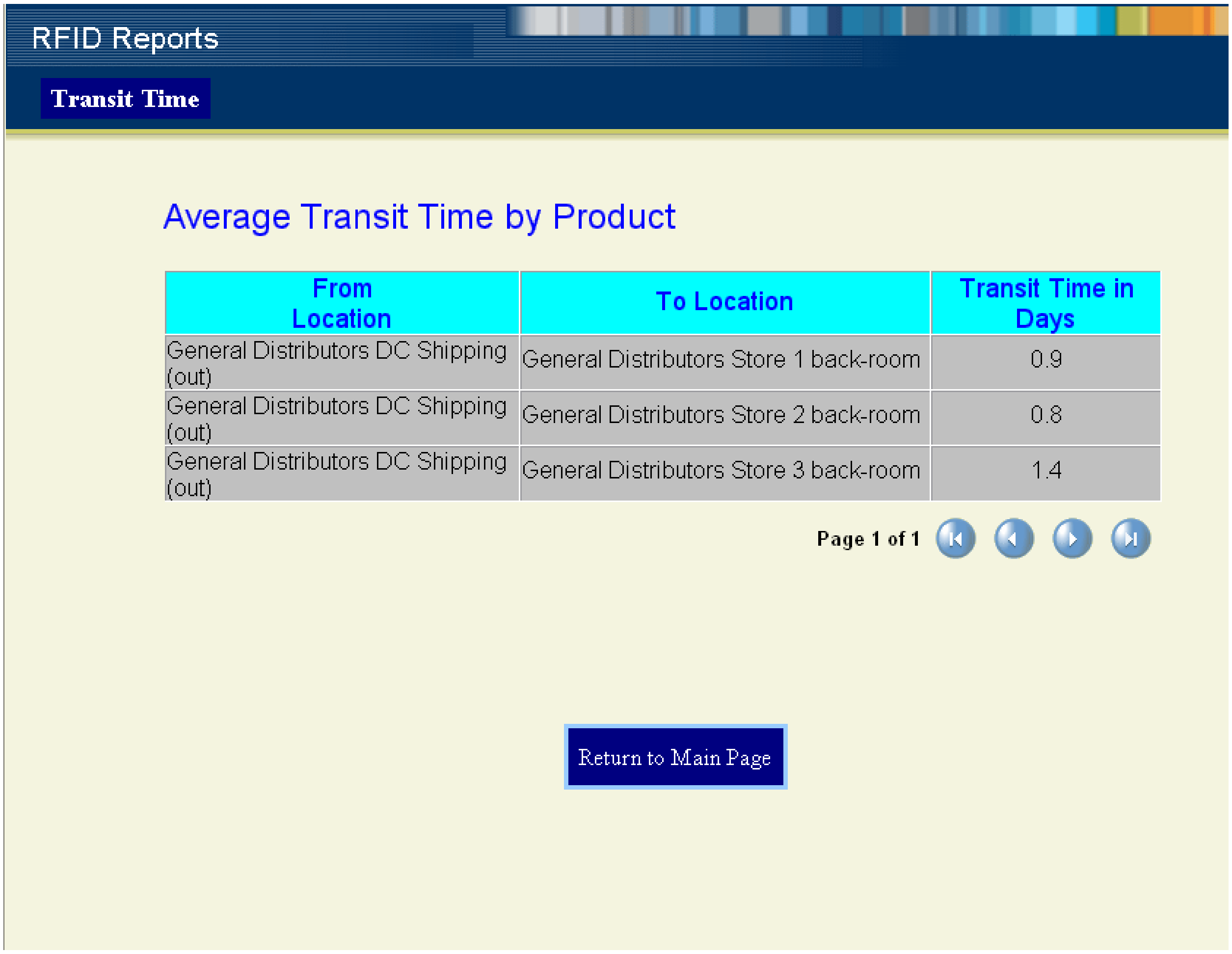
Product ID	Description	Unit of Measurement	Vendor
0000037123500	Butlite 6 Packs	Bottle	Butwiser
0000037126050	Butlite 12 Packs	Bottle	Butwiser
0000037133000	Makealob 6 Packs	Bottle	Butwiser
0000037136250	Makealob 12 Packs	Bottle	Butwiser
0000037145000	O'Dos 6 Packs	Bottle	Butwiser
0000037147050	O'Dos 12 Packs	Bottle	Butwiser
0000038045678	Koca-Kola 6 Packs	Bottle	Koka Kola
0000038045680	Koca-Kola 12 Packs	Bottle	Koka Kola
0000038045200	Spring 6 Packs	Bottle	Koka Kola
0000038045250	Spring 12 Packs	Bottle	Koka Kola

If the product list exceeds one page, you may navigate it by clicking the buttons at the bottom of the list.

2 From the product list, select the product you wish to view by clicking the appropriate **Product ID** link.

The Transit Time report appears.

Figure 102 Transit Time Report



Average Transit Time by Product

From Location	To Location	Transit Time in Days
General Distributors DC Shipping (out)	General Distributors Store 1 back-room	0.9
General Distributors DC Shipping (out)	General Distributors Store 2 back-room	0.8
General Distributors DC Shipping (out)	General Distributors Store 3 back-room	1.4

Page 1 of 1

If the Transit Time report exceeds one page, you may navigate it by clicking the buttons at the bottom of the report.

6.6 EPC Trace Report

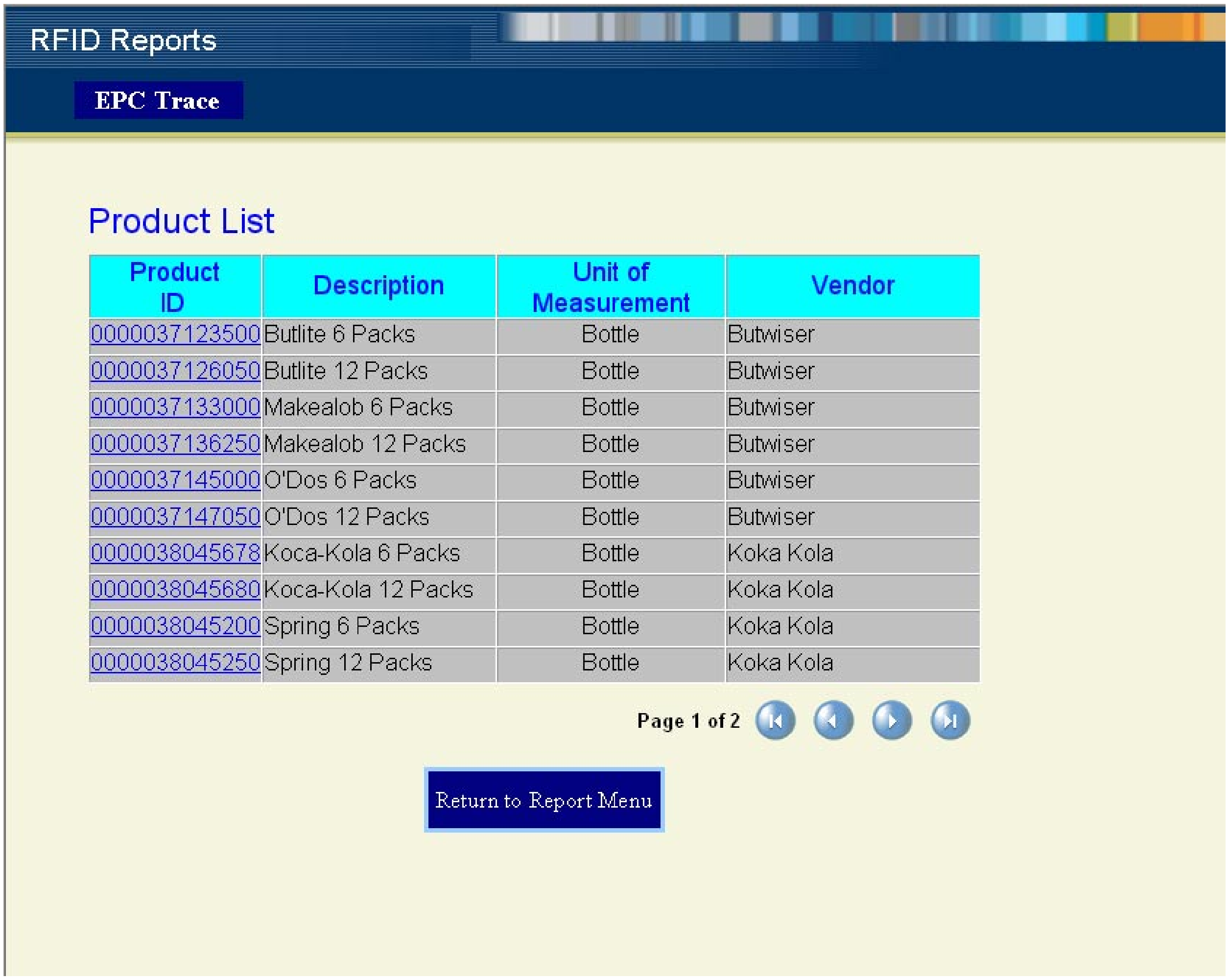
The EPC Trace report shows the EPC trace information for a given product based on specific EPC tag observations.

To view the EPC Trace Report

1 From the RFID Reports page, click **EPC Trace**.

The product list appears.

Figure 103 EPC Trace Product List



Product List

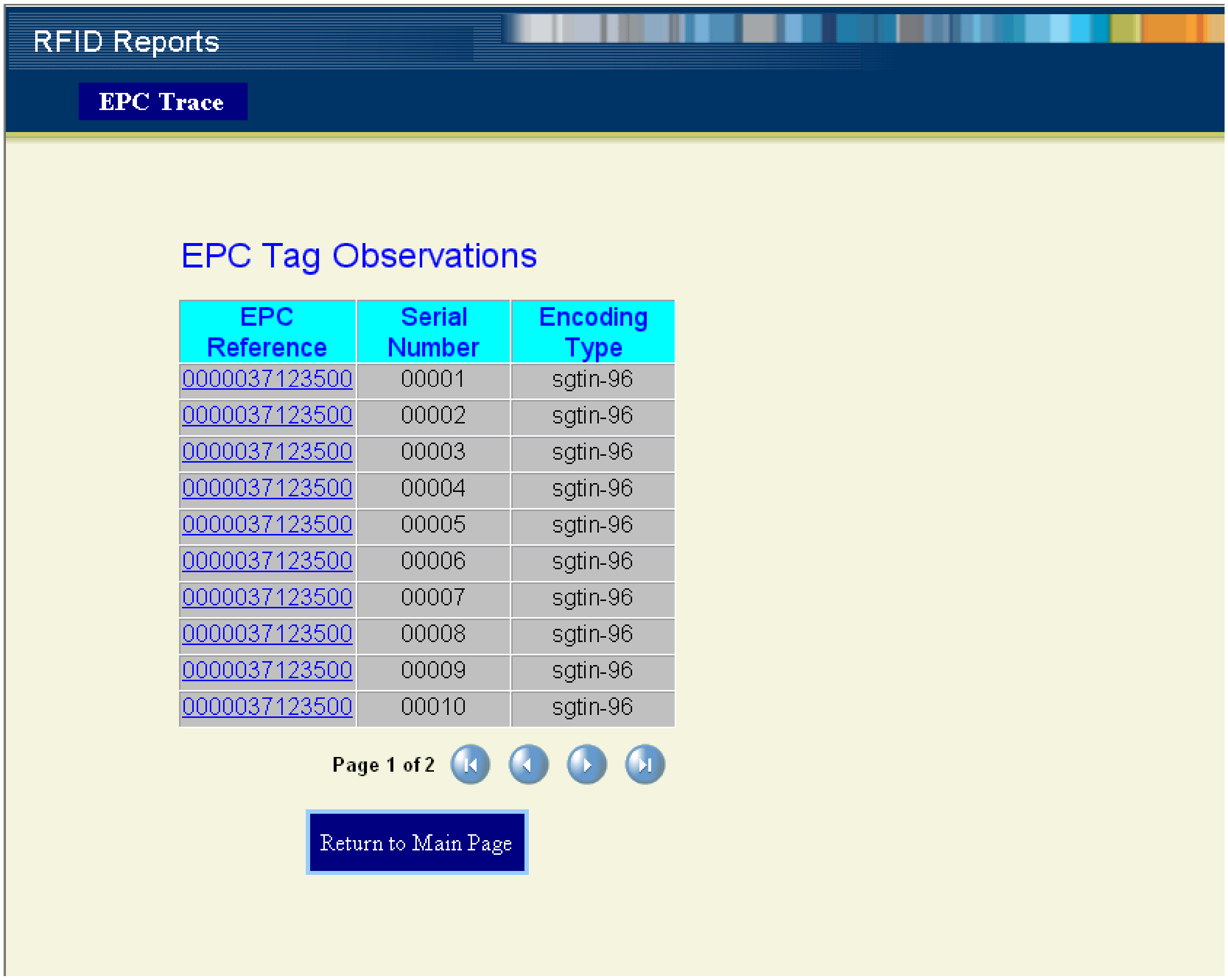
Product ID	Description	Unit of Measurement	Vendor
0000037123500	Butlite 6 Packs	Bottle	Butwiser
0000037126050	Butlite 12 Packs	Bottle	Butwiser
0000037133000	Makealob 6 Packs	Bottle	Butwiser
0000037136250	Makealob 12 Packs	Bottle	Butwiser
0000037145000	O'Dos 6 Packs	Bottle	Butwiser
0000037147050	O'Dos 12 Packs	Bottle	Butwiser
0000038045678	Koca-Kola 6 Packs	Bottle	Koka Kola
0000038045680	Koca-Kola 12 Packs	Bottle	Koka Kola
0000038045200	Spring 6 Packs	Bottle	Koka Kola
0000038045250	Spring 12 Packs	Bottle	Koka Kola

If the product list exceeds one page, you may navigate it by clicking the buttons at the bottom of the list.

2 From the product list, select the product you wish to view by clicking the appropriate **Product ID** link.

The EPC Tag Observations appear.

Figure 104 EPC Trace Tag Observations List



EPC Tag Observations

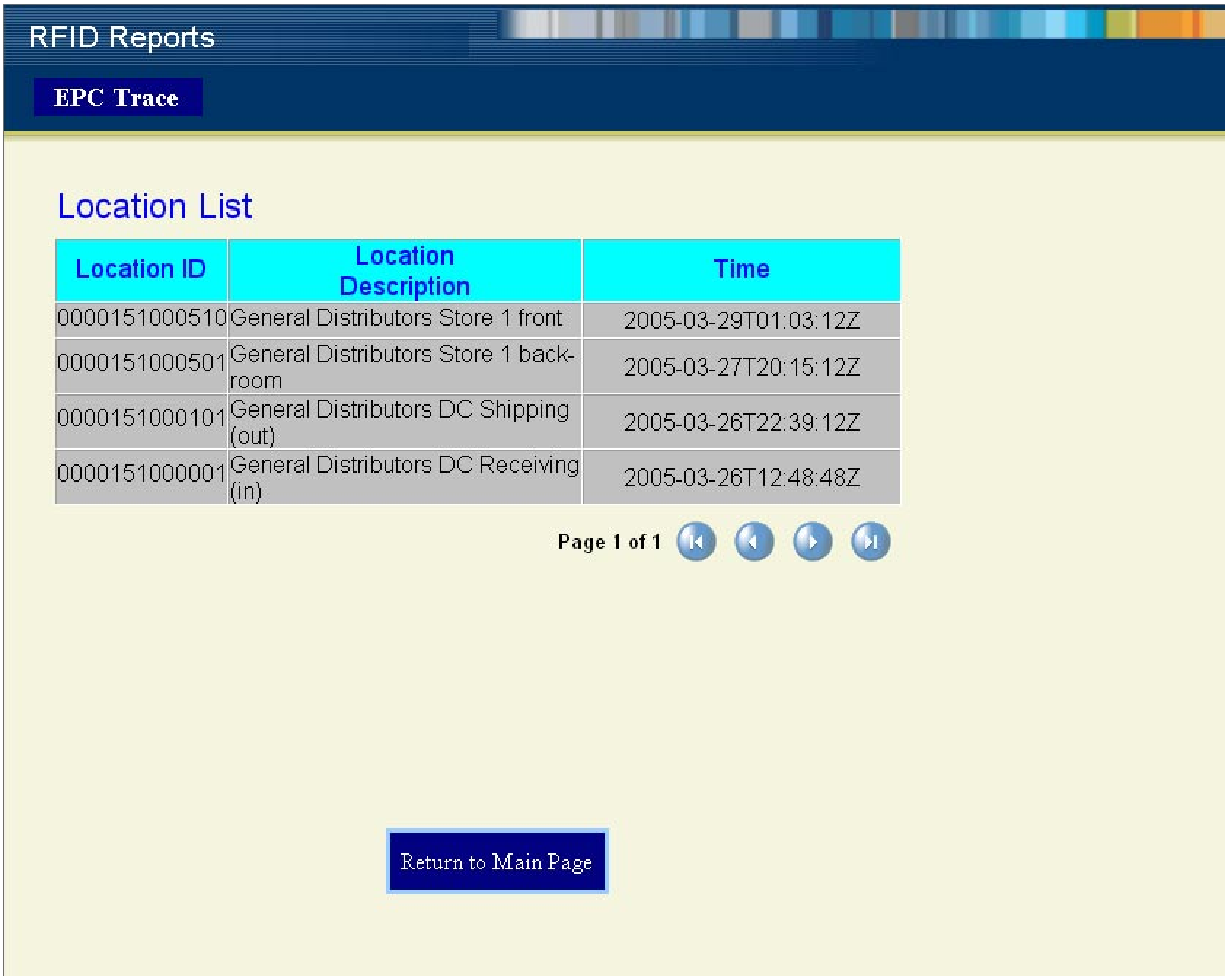
EPC Reference	Serial Number	Encoding Type
0000037123500	00001	sgtin-96
0000037123500	00002	sgtin-96
0000037123500	00003	sgtin-96
0000037123500	00004	sgtin-96
0000037123500	00005	sgtin-96
0000037123500	00006	sgtin-96
0000037123500	00007	sgtin-96
0000037123500	00008	sgtin-96
0000037123500	00009	sgtin-96
0000037123500	00010	sgtin-96

If the EPC Tag Observations list exceeds one page, you may navigate it by clicking the buttons at the bottom of the list.

3 Select the EPC Reference you wish to view by clicking the appropriate **EPC Reference** link.

The EPC Trace report appears.

Figure 105 EPC Trace Report



Location List

Location ID	Location Description	Time
0000151000510	General Distributors Store 1 front	2005-03-29T01:03:12Z
0000151000501	General Distributors Store 1 back-room	2005-03-27T20:15:12Z
0000151000101	General Distributors DC Shipping (out)	2005-03-26T22:39:12Z
0000151000001	General Distributors DC Receiving (in)	2005-03-26T12:48:48Z

If the Trace report exceeds one page, you may navigate it by clicking the buttons at the bottom of the report.

6.7 PO Reconciliation by Item Report

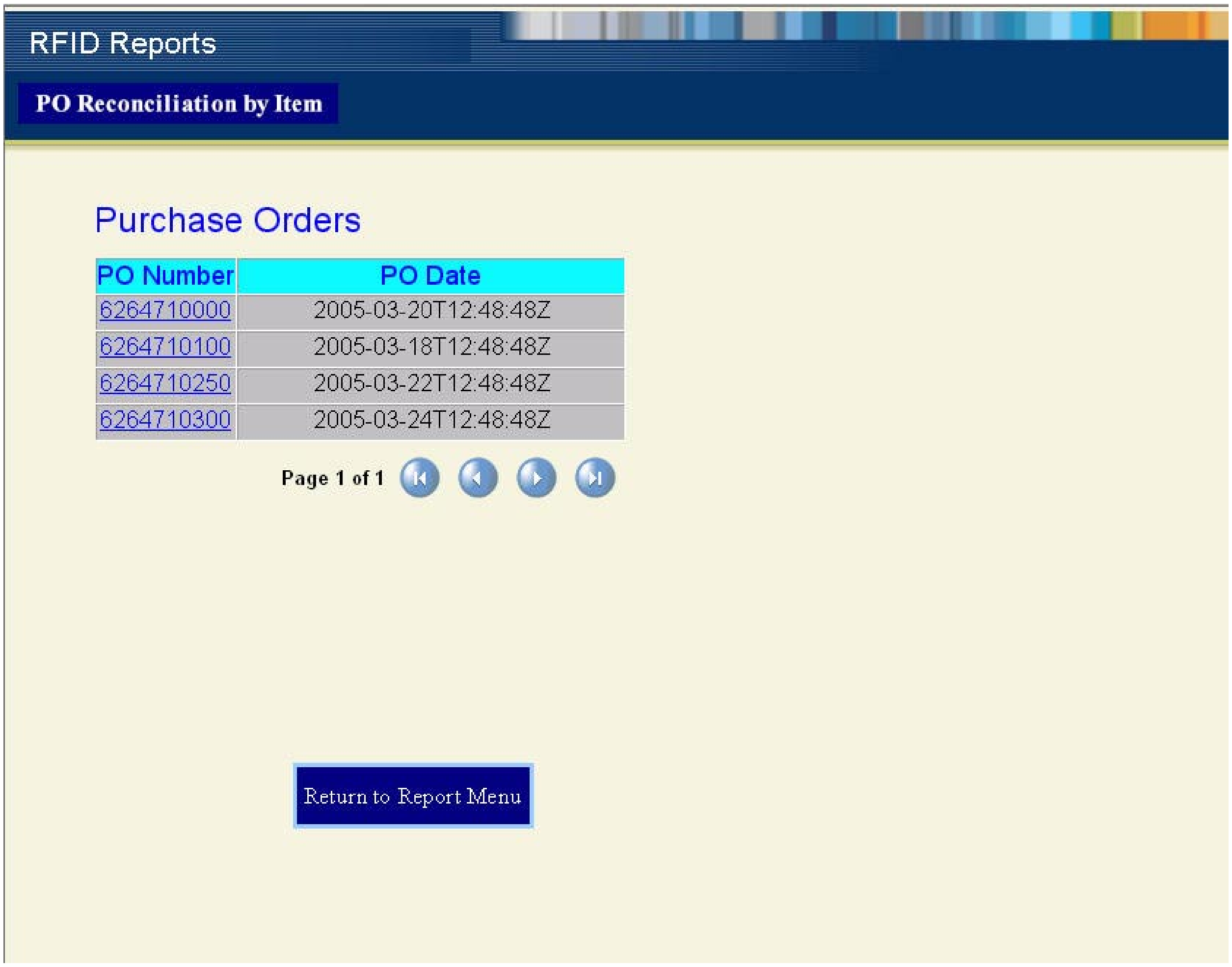
The PO Reconciliation by Item report shows detailed information for purchase orders comparing the quantity ordered, shipped, and received.

To view the PO Reconciliation by Item Report

1 From the RFID Reports page, click **PO Reconciliation by Item**.

The purchase orders list appears.

Figure 106 PO Reconciliation by Item Purchase Order List



Purchase Orders

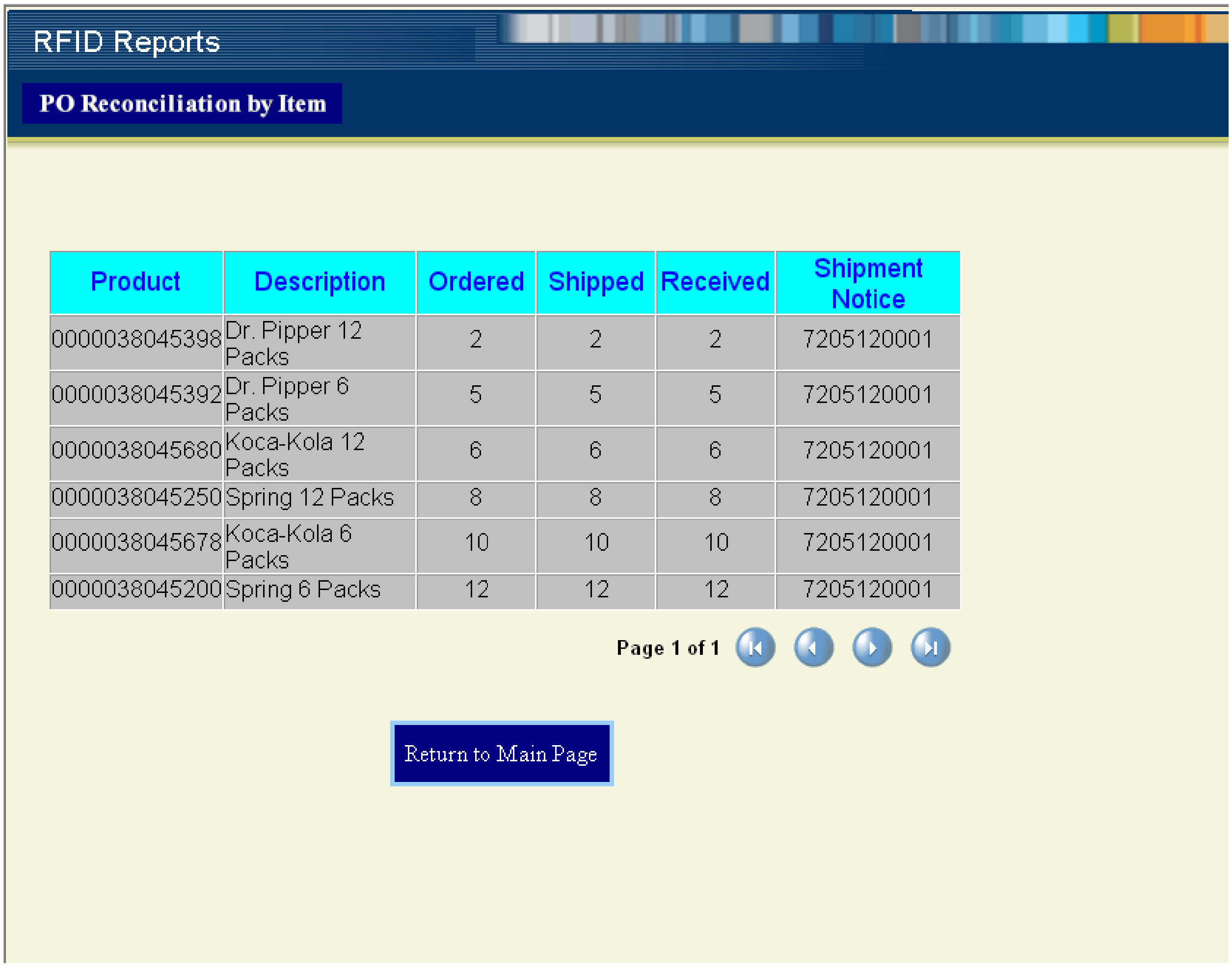
PO Number	PO Date
6264710000	2005-03-20T12:48:48Z
6264710100	2005-03-18T12:48:48Z
6264710250	2005-03-22T12:48:48Z
6264710300	2005-03-24T12:48:48Z

If the purchase order list exceeds one page, you may navigate it by clicking the buttons at the bottom of the list.

2 From the purchase order list, select the purchase order you wish to view by clicking the appropriate **PO Number** link.

The PO Reconciliation by Item report appears.

Figure 107 PO Reconciliation by Item Report



Product	Description	Ordered	Shipped	Received	Shipment Notice
0000038045398	Dr. Pipper 12 Packs	2	2	2	7205120001
0000038045392	Dr. Pipper 6 Packs	5	5	5	7205120001
0000038045680	Koca-Kola 12 Packs	6	6	6	7205120001
0000038045250	Spring 12 Packs	8	8	8	7205120001
0000038045678	Koca-Kola 6 Packs	10	10	10	7205120001
0000038045200	Spring 6 Packs	12	12	12	7205120001

If the PO Reconciliation by Item report exceeds one page, you may navigate it by clicking the buttons at the bottom of the report.

6.8 PO Reconciliation by SSCC Report

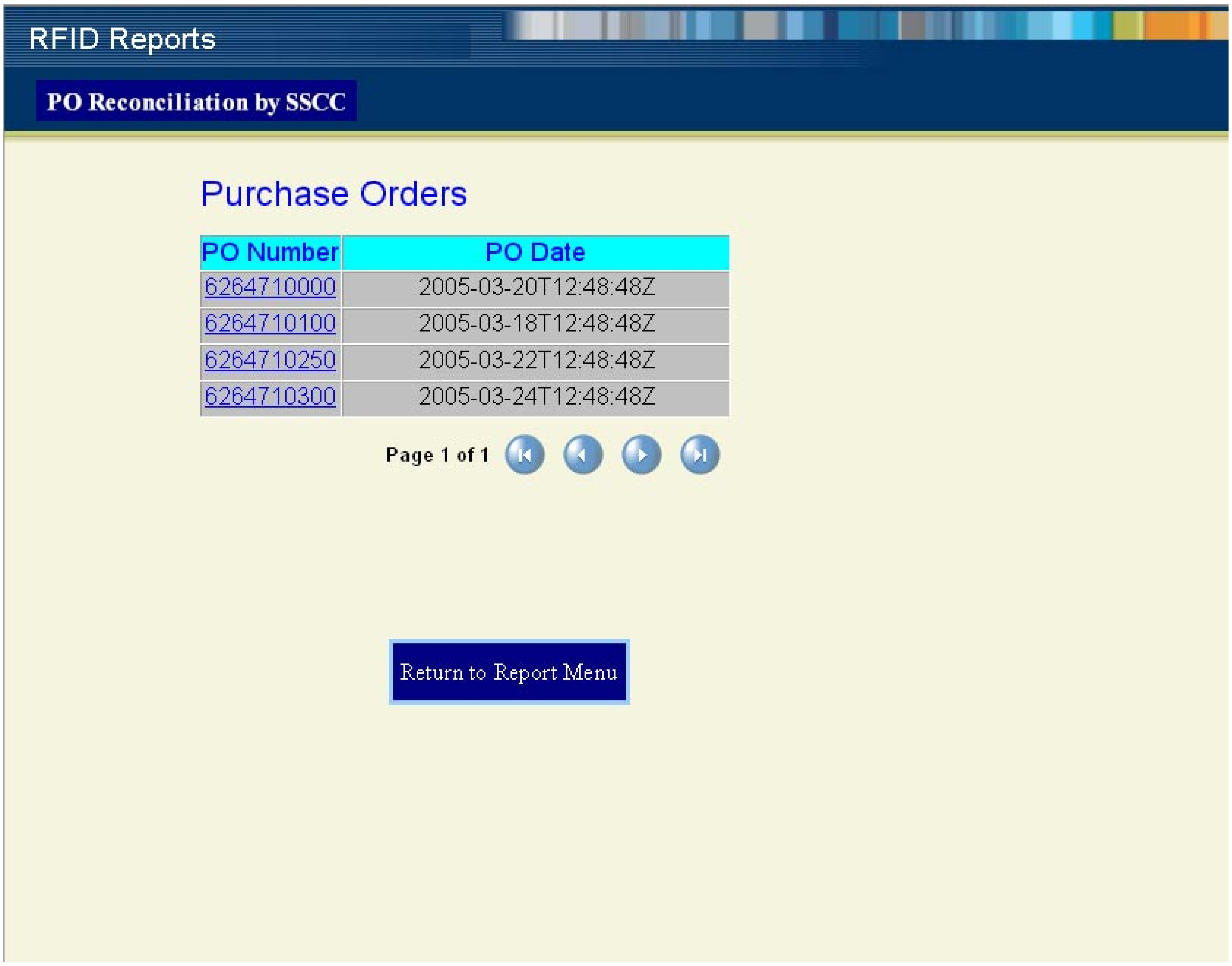
The PO Reconciliation by SSCC report shows detailed information for purchase orders by serial shipping container code (SSCC).

To view the PO Reconciliation by SSCC Report

1 From the RFID Reports page, click **PO Reconciliation by SSCC**.

The purchase orders list appears.

Figure 108 PO Reconciliation by SSCC Purchase Order List



Purchase Orders

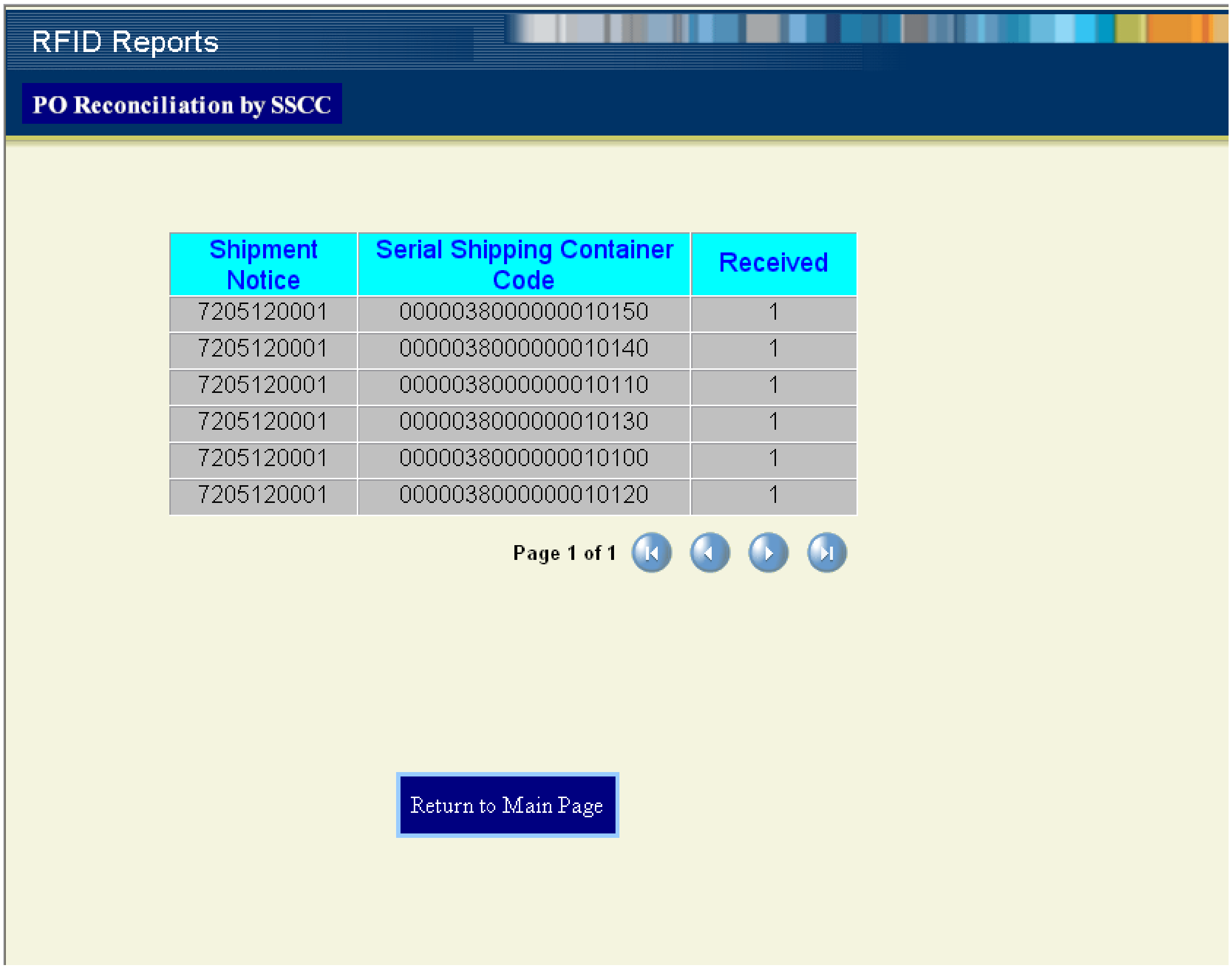
PO Number	PO Date
6264710000	2005-03-20T12:48:48Z
6264710100	2005-03-18T12:48:48Z
6264710250	2005-03-22T12:48:48Z
6264710300	2005-03-24T12:48:48Z

If the purchase order list exceeds one page, you may navigate it by clicking the buttons at the bottom of the list.

2 From the purchase order list, select the purchase order you wish to view by clicking the appropriate **PO Number** link.

The PO Reconciliation by SSCC report appears.

Figure 109 PO Reconciliation by SSCC Report



Shipment Notice	Serial Shipping Container Code	Received
7205120001	00000380000000010150	1
7205120001	00000380000000010140	1
7205120001	00000380000000010110	1
7205120001	00000380000000010130	1
7205120001	00000380000000010100	1
7205120001	00000380000000010120	1

If the PO Reconciliation by SSCC report exceeds one page, you may navigate it by clicking the buttons at the bottom of the report.

6.9 Using the Worklist Manager Alerts

The Worklist Manager shows the alerts for your RFID application. As the user of the Worklist Manager, you have the option of escalating, executing, or marking the alert as complete.

In order to access the Worklist Manager, you must have your LDAP application running. For complete details on configuring LDAP, see **"LDAP Configuration"** in the *eInsight Business Process Manager User's Guide*. If you encounter any problems in configuring LDAP, contact SeeBeyond's technical support at **http://www.seebeyond.com**.

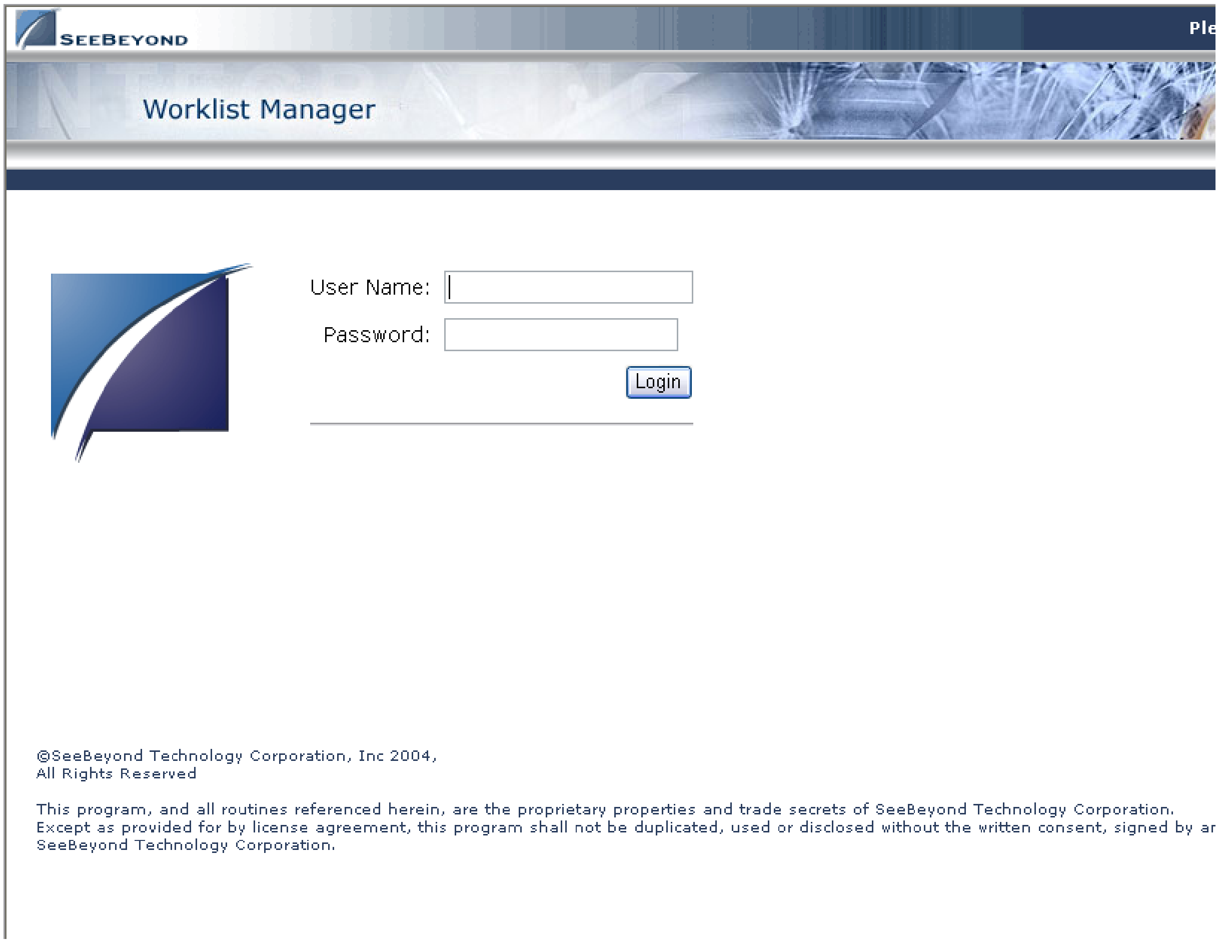
You can access the Worklist Manager with a web browser by entering:

```
http://<hostname>:<port>/wlm
```

- <hostname>: The system where your Repository is running.
- <port>: The port number to access your Repository.

The following login page appears.

Figure 110 Worklist Manager Login

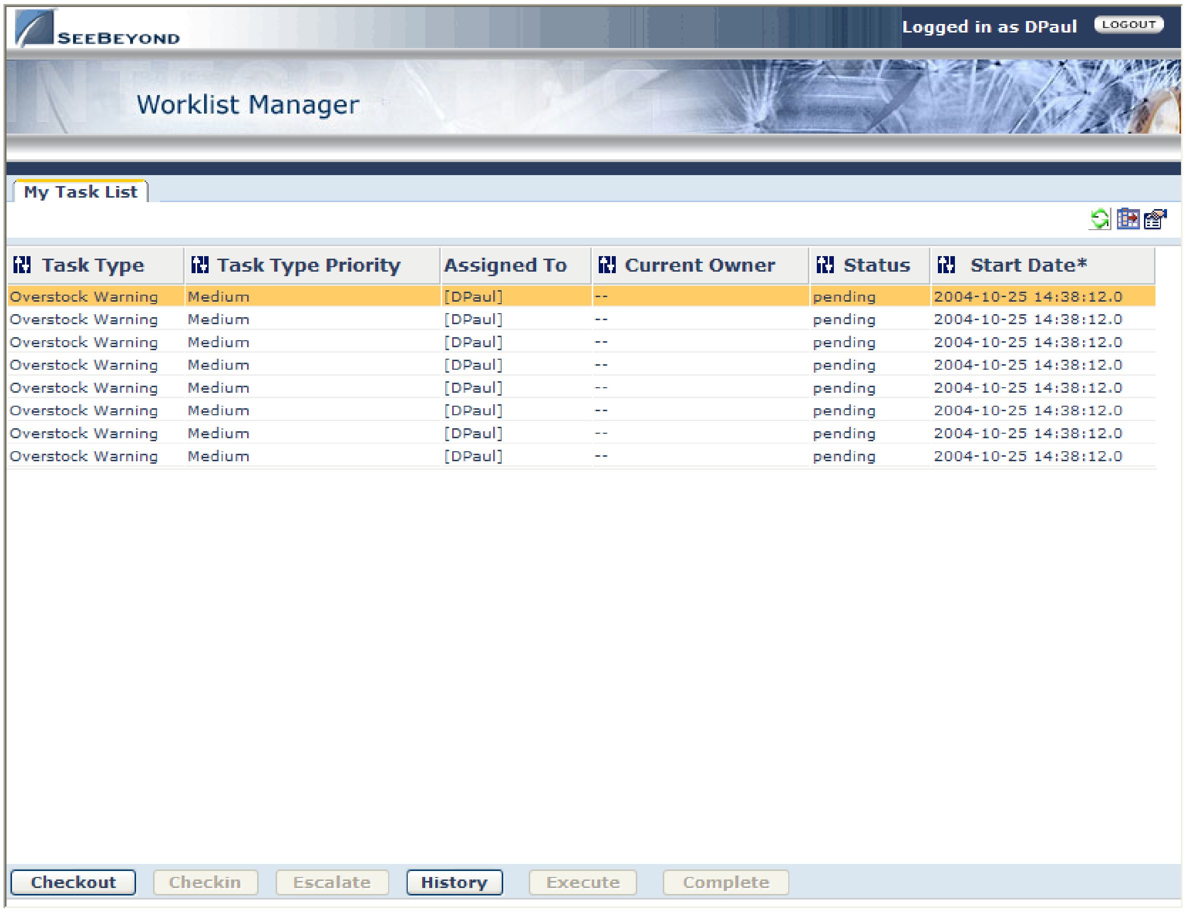


©SeeBeyond Technology Corporation, Inc 2004,
All Rights Reserved

This program, and all routines referenced herein, are the proprietary properties and trade secrets of SeeBeyond Technology Corporation. Except as provided for by license agreement, this program shall not be duplicated, used or disclosed without the written consent, signed by ar SeeBeyond Technology Corporation.

Once you enter the assigned user name **dpaul** and the password **pass**, the Worklist Manager task view appears.

Figure 111 Worklist Manager

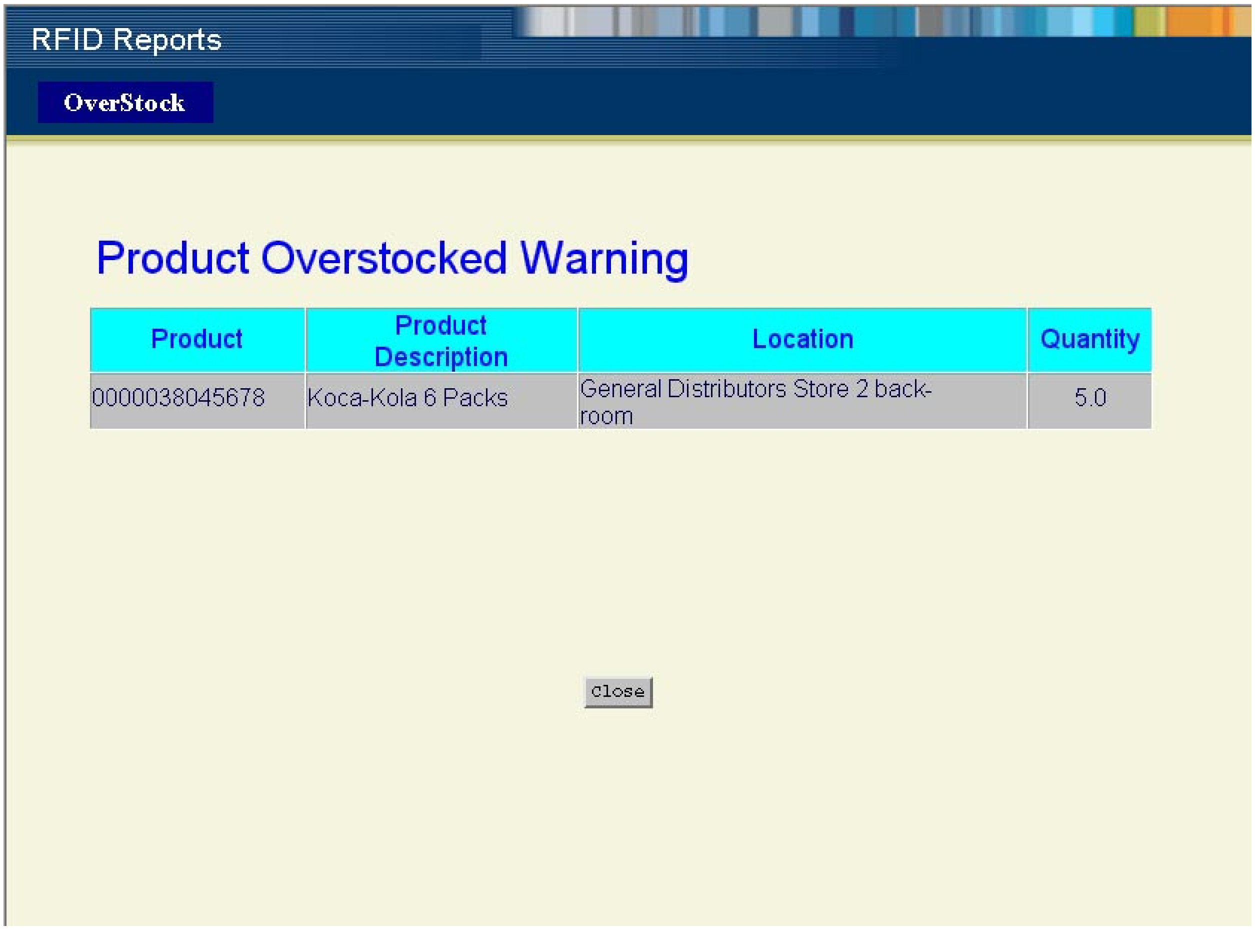


Task Type	Task Type Priority	Assigned To	Current Owner	Status	Start Date*
Overstock Warning	Medium	[DPaul]	--	pending	2004-10-25 14:38:12.0
Overstock Warning	Medium	[DPaul]	--	pending	2004-10-25 14:38:12.0
Overstock Warning	Medium	[DPaul]	--	pending	2004-10-25 14:38:12.0
Overstock Warning	Medium	[DPaul]	--	pending	2004-10-25 14:38:12.0
Overstock Warning	Medium	[DPaul]	--	pending	2004-10-25 14:38:12.0
Overstock Warning	Medium	[DPaul]	--	pending	2004-10-25 14:38:12.0
Overstock Warning	Medium	[DPaul]	--	pending	2004-10-25 14:38:12.0
Overstock Warning	Medium	[DPaul]	--	pending	2004-10-25 14:38:12.0

Here is an explanation of the options you will see:

- **Checkout/Checkin** views and closes the source of the alert, and is necessary to ensure that more than one person is not working on the same alert.
- **Escalate** sends the alert to the user's manager.
- **History** provides a record of all past information about the alert.
- **Execute** opens the report for the alert (see Figure 112 on page 121).
- **Complete** commits the changes. You must select **Complete** before leaving the Worklist Manager or your changes will be lost.

Figure 112 Product Overstocked Warning



Product Overstocked Warning

Product	Product Description	Location	Quantity
0000038045678	Koca-Kola 6 Packs	General Distributors Store 2 back-room	5.0

After viewing the report, click **Close** to return to the Worklist Manager.

For more information on using the Worklist Manager, see the *eInsight Business Process Manager User's Guide*.

Viewing sample reports

To view the sample data in the Worklist Manager without processing any RFID tag records, you can load sample data into your database. See **"Populating the Tables with Sample Data" on page 88** for more information.

Additionally, you must create a file called **TATOverstockAlert.txt** and copy it to the **C:\test** directory. This file should contain the number of days of overstock (similar to the overstock value used by the **"OverStock Report" on page 108**).

Chapter 7

Using the Single WebService Interface for Sample EPC IS

What's in This Chapter

- '"About Web Services Operations" on page 122
- "Configuring Web Services Operations" on page 122
- "Running a Sample Web Services Operation" on page 124

7.1 About Web Services Operations

With Web Services becoming a standard in EPC IS, it is also becoming increasingly popular as an interface. It provides an additional option to access and maintain data stored in the sample EPC IS database. By using the available tools in ICAN Suite 5.0, the following section describes how a sample business process which incorporates Java Collaboration can be utilized to achieve data access through Web Services.

7.2 Configuring Web Services Operations

Since EPC IS is required to support multiple operations, an `operation` element is used to indicate the name of the operation. An operation can take parameters, thus the `parameter` element has multiple input values. To allow an option for the service implementation to process a parameter value based on name instead of an index, an optional `name` element is added in the parameter type. Following is the WSDL description of the operation name and parameters:

```
<complexType name="ProcessRequestType">
    <sequence>
        <element name="Operation" type="string"/>
        <element name="Parameter" type="ParameterType"
minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
</complexType>
<complexType name="ParameterType">
    <sequence>
        <element name="Name" type="string" minOccurs="0"/>
        <element name="Value" type="anyType"/>
    </sequence>
```

```
</complexType>
```

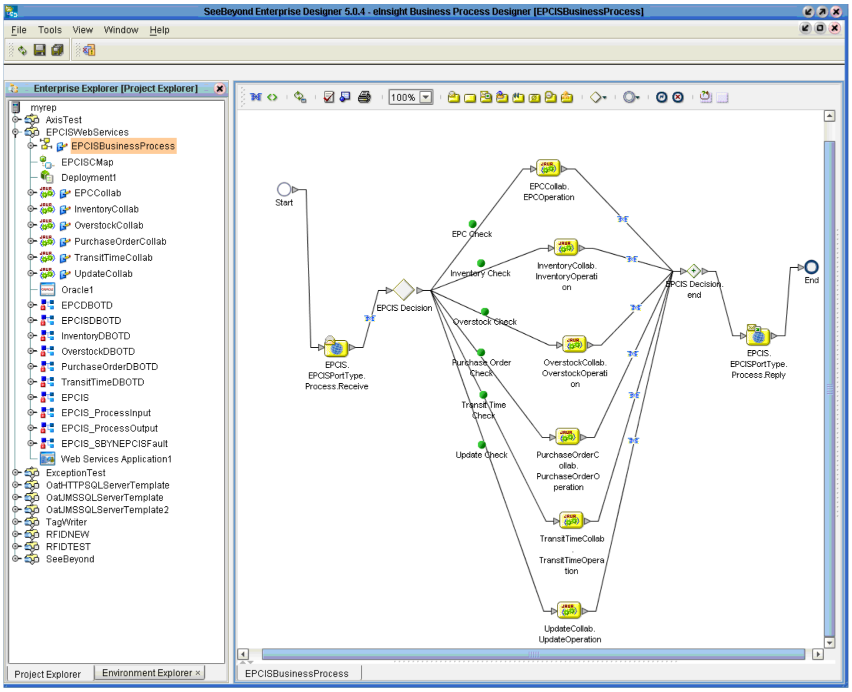
An operation can also return an output. To give an option for the client implementation to dynamically discover the name and type of a particular value, a `fieldmetadata` element is added. In addition, more than one value can be returned. Thus the output values need to return as records, as described by the following WSDL code segment:

```
<complexType name="ProcessResponseType">
    <sequence>
        <element name="FieldMetadata" type="FieldMetadataType"
maxOccurs="unbounded"/>
        <element name="Record" type="RecordType" minOccurs="0"
maxOccurs="unbounded"/>
    </sequence>
</complexType>
<complexType name="FieldMetadataType">
    <sequence>
        <element name="FieldName" type="string"/>
        <element name="FieldType" type="string"/>
    </sequence>
</complexType>
<complexType name="RecordType">
    <sequence>
        <element name="FieldValue" type="anyType"
maxOccurs="unbounded"/>
    </sequence>
</complexType>
```

To allow maximum flexibility, these types are creaded in a separate XSD from the WSDL. Current implementation of BPEL does not support repeating nodes of repeating nodes. As a result, the business logic is implemented in Java Collaboration. Services are grouped into different Java Collaborations depending on its function, thus enabling easier maintenance.

The Business Process Diagram below displays how Java Collaborations are being linked to the generic WSDL OTD service.

Figure 113 Web Services Java Collaborations



7.3 Running a Sample Web Services Operation

To test the WebService feature in the RFID eWay by running a sample operation, use any generic SOAP client such as SOAPClient4xg. Before testing the WebService, make sure to create all the tables for the sample EPC IS application and populate tables with sample data. See **“Populating the Tables with Sample Data” on page 88** for more information.

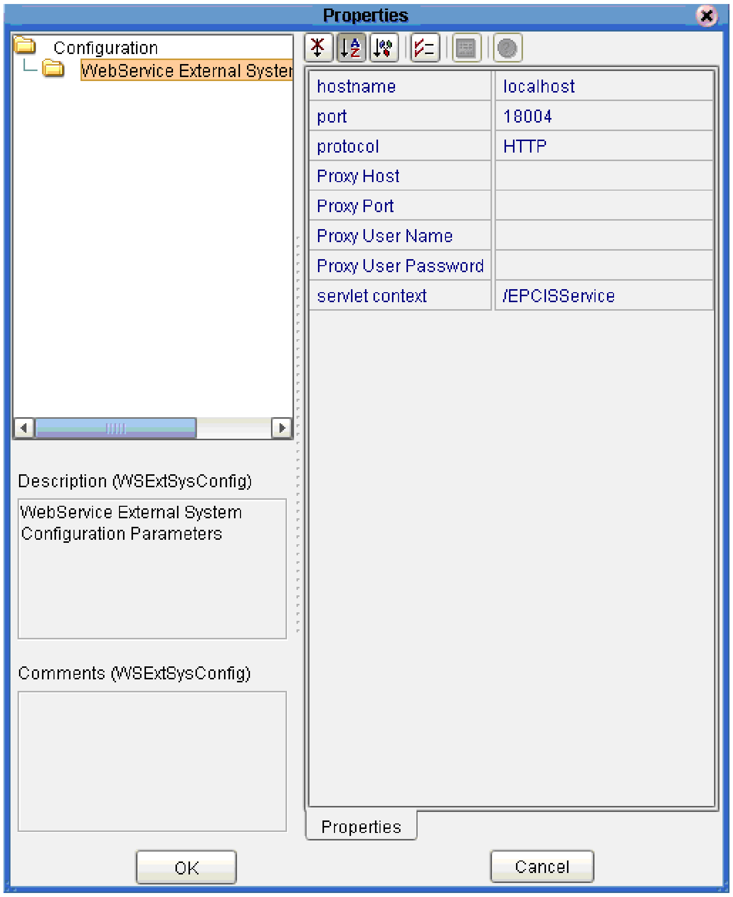
To run SOAPClient4xg, three inputs are needed. The first input is the name of the ICAN Suite WebService, **http://hostname:port/servletcontext**, which can be obtained from the properties window of the WebService in **Enterprise Designer.**

1 In the Enterprise Explorer pane, click the Environment Explorer tab.

2 Right-click the WebService section for your project, and then click **Properties**.

The **Properties** window appears.

Figure 114 Web Services Properties



hostname	localhost
port	18004
protocol	HTTP
Proxy Host	
Proxy Port	
Proxy User Name	
Proxy User Password	
servlet context	/EPCISService

Description (WSExtSysConfig)

WebService External System Configuration Parameters

Comments (WSExtSysConfig)

An example of the second input for SOAPClient4xg is the following (this example executes the Inventory Collaboration illustrated in **Figure 113 on page 124**):

```
<SOAP-ENV:Envelope
   xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
   xmlns:xsd="http://www.w3.org/1999/XMLSchema"
   xmlns:ns1="http://epcis.webservice.seebeyond"
   xmlns:ns2="http://epcis.webservice.seebeyond/1.0/types"
>
   <SOAP-ENV:Body>
      <ns2:Process>
         <ns2:Operation>QueryInventoryByLocation</ns2:Operation>
         <ns2:Parameter>
         <ns2:Value>0000151000006</ns2:Value>
      </ns2:Parameter>
      </ns2:Process>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The highlighted value is the location identifier in the Sample EPC IS application. The XML data is based on the following XSD schema which can also be found in the the directory **..\edesigner\usrdir\modules\ext\rfidadapter\templates\Oracle\WebServices**:

```
<?xml version="1.0"?>
<schema targetNamespace="http://epcis.webservice.seebeyond/1.0/types"
    xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:seebeyond="http://www.seebeyond.com/2004/xml"
    elementFormDefault="qualified">
        <annotation>
            <documentation>
                <seebeyond:copyright>Copyright 2004 SeeBeyond, All
Rights Reserved.</seebeyond:copyright>
            </documentation>
        </annotation>
        <element name="Process" type="ProcessRequestType"/>
    <complexType name="ProcessRequestType">
        <sequence>
            <element name="Operation" type="string"/>
            <element name="Parameter" type="ParameterType"
minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
    </complexType>
    <complexType name="ParameterType">
        <sequence>
            <element name="Name" type="string" minOccurs="0"/>
            <element name="Value" type="anyType"/>
        </sequence>
    </complexType>
    <element name="ProcessResponse" type="ProcessResponseType"/>
    <complexType name="ProcessResponseType">
        <sequence>
            <element name="FieldMetadata" type="FieldMetadataType"
maxOccurs="unbounded"/>
            <element name="Record" type="RecordType" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
    </complexType>
    <complexType name="FieldMetadataType">
        <sequence>
            <element name="FieldName" type="string"/>
            <element name="FieldType" type="string"/>
        </sequence>
    </complexType>
    <complexType name="RecordType">
        <sequence>
            <element name="FieldValue" type="anyType"
maxOccurs="unbounded"/>
        </sequence>
    </complexType>
    <element name="ProcessFault" type="string" minOccurs="0"/>
</schema>
```

The Web Service is based on the following WSDL which can be found in the same directory listed above:

```
<definitions targetNamespace="http://epcis.webservice.seebeyond"
xmlns:sbynpx="http://bpel.seebeyond.com/hawaii/5.0/privateExtension/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
    xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
    xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:s0="http://epcis.webservice.seebeyond"
    xmlns:s1="http://epcis.webservice.seebeyond/1.0/types"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
    <types>
        <schema xmlns="http://www.w3.org/2001/XMLSchema">
```

```
            <import namespace="http://epcis.webservice.seebeyond/
1.0/types" schemaLocation="./EPCIS.xsd"/>
        </schema>
    </types>
    <message name="SBYNEPCISFault">
        <part name="Fault" element="s1:ProcessFault"/>
    </message>
    <message name="ProcessInput">
        <part name="InputParameters" element="s1:Process"/>
    </message>
    <message name="ProcessOutput">
        <part name="OutputParameters" element="s1:ProcessResponse"/>
    </message>
    <portType name="EPCISPortType">
        <operation name="Process">
            <documentation>Process EPCIS Requests.</documentation>
            <input message="s0:ProcessInput"/>
            <output message="s0:ProcessOutput"/>
            <fault name="ProcessFault" message="s0:SBYNEPCISFault"/>
        </operation>
        </portType>
        <binding name="EPCISBinding" type="s0:EPCISPortType">
            <soap:binding style="document" transport="http://
schemas.xmlsoap.org/soap/http"/>
            <operation name="Process">
                <soap:operation soapAction="http://
epcis.webservice.seebeyond/EPCISPortType/Process" style="document"/>
                <input>
                    <soap:body use="literal"/>
                </input>
                <output>
                    <soap:body use="literal"/>
                </output>
                <fault name="ProcessFault">
                    <soap:body use="literal"/>
                </fault>
            </operation>
        </binding>
        <service name="EPCISService">
            <port name="EPCISBindingService"
binding="s0:EPCISBinding">
                <soap:address location="http://jlong-
d600.stc.com:18004/EPCISService"/>
            </port>
        </service>
</definitions>
```

The third input for SOAPClient4xg is **http://epcis.webservice.seebeyond/EPCISPortType/Process**. After running the program, the output appears as follows:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/
envelope/" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:s1="http://
epcis.webservice.seebeyond/1.0/types" xmlns:sbynpx="http://
bpel.seebeyond.com/hawaii/5.0/privateExtension/" xmlns:seebeyond
="http://www.seebeyond.com/2004/xml" xmlns:si0="http://
epcis.webservice.seebeyond" xmlns:soap="http://schemas.xmlsoap.org/
wsdl/soap/" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"><SOAP-
ENV:Header/><SOAP-ENV:Body><s1:ProcessResponse>
    <s1:FieldMetadata>
        <s1:FieldName>COMPANY_REFERENCE</s1:FieldName>
        <s1:FieldType>string</s1:FieldType>
    </s1:FieldMetadata>
    <s1:FieldMetadata>
```

```
        <s1:FieldName>DESCRIPTION</s1:FieldName>
        <s1:FieldType>string</s1:FieldType>
    </s1:FieldMetadata>
    <s1:FieldMetadata>
        <s1:FieldName>QUANTITY</s1:FieldName>
        <s1:FieldType>decimal</s1:FieldType>
    </s1:FieldMetadata>
    <s1:Record>
        <s1:FieldValue>0000037123500</s1:FieldValue>
        <s1:FieldValue>Butlite 6 Packs</s1:FieldValue>
        <s1:FieldValue>2</s1:FieldValue>
    </s1:Record>
    <s1:Record>
        <s1:FieldValue>0000038045678</s1:FieldValue>
        <s1:FieldValue>Koca-Kola 6 Packs</s1:FieldValue>
        <s1:FieldValue>5</s1:FieldValue>
    </s1:Record>
</s1:ProcessResponse></SOAP-ENV:Body></SOAP-ENV:Envelope>
```

Chapter 8

Using the RFID eWay EDI Option

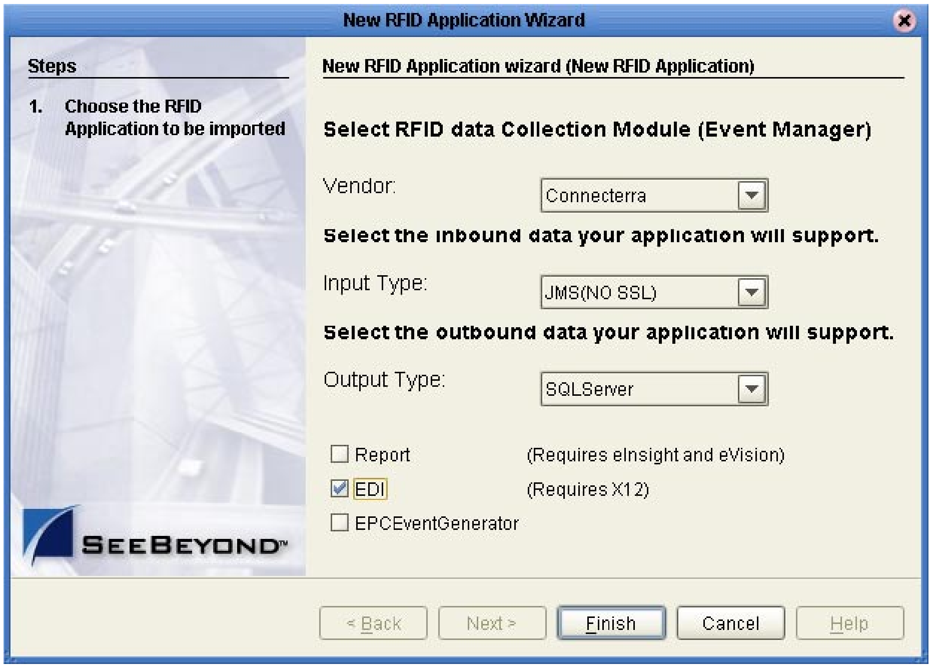
What's in This Chapter

- "Activating EDI for the Sample EPC IS Application" on page 129

8.1 Activating EDI for the Sample EPC IS Application

When you create a new RFID application, you choose to add the EDI option to the Project or not. When you select the EDI option (see the following figure), the wizard includes the EDI Project in your RFID application.

Figure 115 EDI Option



New RFID Application Wizard

Steps

1. Choose the RFID Application to be imported

New RFID Application wizard (New RFID Application)

Select RFID data Collection Module (Event Manager)

Vendor: Connecterra

Select the inbound data your application will support.

Input Type: JMS(NO SSL)

Select the outbound data your application will support.

Output Type: SQLServer

☐ Report (Requires eInsight and eVision)

☑ EDI (Requires X12)

☐ EPCEventGenerator

Configuring the EDI Project requires only a few extra steps beyond the configuration of your RFID Project. When you create the Environment, also add an inbound File External System. When you add the Deployment Profile to the EDI Project, drag each of the components to the appropriate container in the Deployment Profile—drag the

Services to the Integration Server, the File eWays to the File external, and the Oracle eWays to the Oracle external.

Sample input files

Two sample input files are provided with this eWay. These sample files supply the input data used by the File eWay in the EDI sample Project.

These input files are located in **<enterprise_designer_root>\edesigner\usrdir\modules\ext\rfiadapter\samples**.

Index

A

alerts **119**

C

configuring third-party event managers **14–25**
ConnecTerra **8**, **13**
ConnecterraHTTPDB2Template, using **72**
ConnecterraHTTPOracleTemplate, using **69**
ConnecterraHTTPSQLServerTemplate, using **75**
ConnecterraHTTPTigerLogicTemplate, using **78**
ConnecterraJMSDB2Template, using **53**
ConnecterraJMSOracleTemplate, using **50**
ConnecterraJMSSQLServerTemplate, using **56**
ConnecterraJMSTigerLogicTemplate, using **59**
conventions, document **9**

D

document
  conventions **9**
  scope **9**
documents, related **10**

E

EDI **27**, **122**, **124**, **129–130**
  activating **129**
EPC Event Generator **7**, **89–97**
  configure **93**
  logical host **94**
  oracle template, configure **92**
EPC information service **7**
EPC trace report **112**
event managers **7**
  third-party configuration **14–25**

H

HP-UX **12**
HTTPJMSQueueTemplate, using **65**

I

IBM AIX **12**
installation **12–25**
  eWay product files **14**
inventory report **106**

J

JMS binding utility **23**

L

Linux
  Red Hat **12**

O

Observation_Log Database **84**
  archiving transactions **86**
  creating tables **85**
  sample data **88**
  sample EPC IS application **87**
  SQL files **84**
operating systems
  HP-UX **12**
  IBM AIX **12**
  Red Hat Linux **12**
  Sun Solaris **12**
  Windows **12**
overstock report **108**

P

platforms. *See* operating systems
PO reconciliation by item report **115**
PO reconciliation by SSCC report **117**
port number **119**
prerequisite products **13**
project template, selecting **27**

R

Red Hat Linux **12**
Reports **27**
reports **7**, **98–118**
  activating **99**
  EPC Trace **112**
  installing **99**
  Inventory **106**
  Overstock **108**
  PO Reconciliation by Item **115**
  PO Reconciliation by SSCC **117**
  Transit Time **110**

requirements **12–13**
  JMS **13**
  system **12**
  third-party **13**
RFID eWay
  about **7**
  configuring **26–83**
  ConnecTerra, working with **8**
  creating projects **26**
  EDI **27**
  installation **12–25**
  key features **7**
  modes of operation **8**
  project templates **27**
  Reports **27**
  Sun RFID Event Manager, working with **8**
RFID eway
  create a new application **26**
RFID, about **6**
RFTagAware **13**

S

sample data **88**
screenshots **10**
selecting a project template **27**
Sun RFID Event Manager **8**, **13**
Sun Solaris **12**
SunRFIDSuiteJMSDB2Template, using **37**
SunRFIDSuiteJMSOracleTemplate, using **29**
SunRFIDSuiteJMSSQLServerTemplate, using **40**
SunRFIDSuiteJMSTigerLogicTemplate, using **43**
system requirements
  external **13**
  ICAN **12**

T

third-party
  event manager configuration **14–25**
  requirements **13**
transit time report **110**

W

WebService Interface **122**
  Configuring **122**
Windows **12**
Worlkist Manager **119**