

**SeeBeyond ICAN Suite™**

# Siebel UAN eWay Intelligent Adapter User's Guide

*Release 5.0.1*



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, eGate, and eWay are the registered trademarks of SeeBeyond Technology Corporation in the United States and select foreign countries; the SeeBeyond logo, e\*Insight, and e\*Xchange 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.

© 2003-2004 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 This work is protected as an unpublished work under the copyright laws.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 20031224111300.

# Contents

---

## Chapter 1

<b>Introducing the Siebel UAN eWay</b>	<b>5</b>
<b>Overview</b>	<b>5</b>
<b>Operation</b>	<b>6</b>
Typical Process Flow	6
<b>XSLT Collaboration</b>	<b>7</b>
XML Message Transformation	7
XML Message Parsing	7
XML Message Validation	8
ID Cross-Reference Mapping	8
Value Cross-Reference Mapping	8
<b>Cross-Reference Database</b>	<b>8</b>
<b>Supported Operating Systems</b>	<b>9</b>
<b>Host System Requirements</b>	<b>9</b>
Logical Host Requirements	9
<b>External System Requirements</b>	<b>9</b>

---

## Chapter 2

<b>Installing the Siebel UAN eWay</b>	<b>10</b>
<b>Overview</b>	<b>10</b>
<b>Installing on Windows Operating Systems</b>	<b>11</b>

---

## Chapter 3

<b>Setting Properties of Siebel UAN eWay</b>	<b>12</b>
<b>Creating and Configuring the Siebel UAN eWay</b>	<b>12</b>
<b>Configuring the eWay Connectivity Map Properties</b>	<b>12</b>
Siebel UAN Outbound eWay Properties	12
eWay Property Settings	14
JDBC Driver Class	14
<b>Configuring the Environment Properties</b>	<b>14</b>
Siebel UAN External System Outbound Properties	14
Environment Property Settings	15
URL	15

## Contents

User Name	16
Password	16

<b>Index</b>	<b>17</b>
--------------	-----------

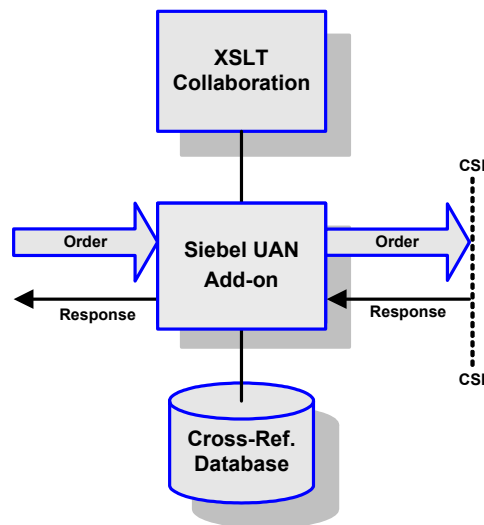
# Introducing the Siebel UAN eWay

This chapter provides a brief introduction to the SeeBeyond Intelligent Adapter for Siebel UAN.

## 1.1 Overview

The Intelligent Adapter for Siebel UAN is a unique eWay component in that its primary functionality is internal to eGate Integrator. Although it connects directly to a cross-reference database, it connects to other applications through other eWays. It is specifically designed to provide the core functionality in a Siebel UAN Integration Server.

**Figure 1** Siebel UAN eWay

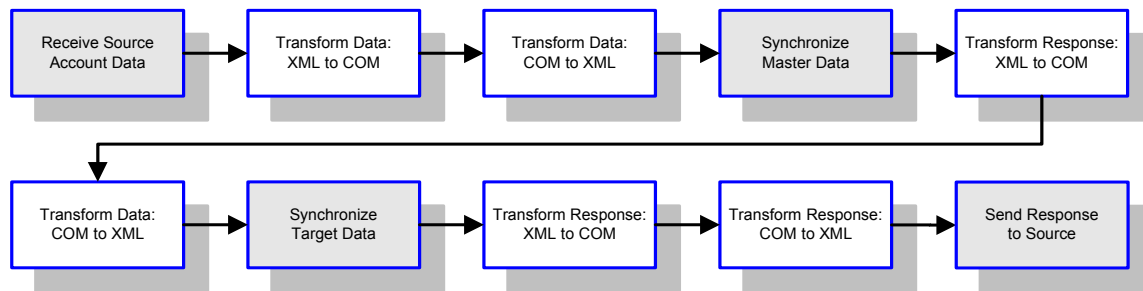


The Intelligent Adapter for Siebel UAN primarily receives messages originating in the Front end applications like Siebel CRM and Back office applications like SAP and Oracle through application specific eWays. It then performs a transformation to the UAN Common Object Model (COM) using an XSLT Collaboration, and provides the result through the Common Services Interface (CSI). Finally, it propagates a response from the target application back to the source application. The Siebel UAN eWay also provides cross-referencing between the unique IDs assigned to messages by the various applications in the integrated system.

## 1.1.1 Operation

A typical example of UAN eWay operation within a UAN project is shown in Figure 2. The example used corresponds to the Sync Account IAP, but other IAP projects are similar in nature. Each transformation from XML to COM, or COM to XML, is contained within a Collaboration performed by the UAN eWay.

**Figure 2** Typical UAN Operation



### Typical Process Flow

- 1 The user of a supported application enters data and clicks the accept button in the source application user interface.
- 2 The source application posts an XML document to the eGate Integrator system, which presents it to the UAN eWay.
- 3 The eGate Integrator system starts the e\*Insight business process.
- 4 The UAN eWay transforms the XML document into the COM format, and adds it to the cross-reference database.
- 5 The UAN eWay transforms the COM data into XML format.
- 6 The SyncAccount process synchronizes the master application using this data.
- 7 The UAN eWay receives the XML response via the related eWay, transforms it into COM format, and adds it to the cross-reference database.
- 8 The UAN eWay transforms the COM data into XML format.
- 9 The SyncAccount process synchronizes the target application using this data.
- 10 The target eWay retrieves the target application response and presents it to the UAN eWay.
- 11 The UAN eWay transforms the response into COM format, and adds it to the cross-reference database.
- 12 The UAN eWay then transforms the COM response in the cross-reference database to XML format, and returns it to the source application via the source eWay.

## 1.1.2 XSLT Collaboration

The primary functionality of the XSLT Collaboration is provided by the Object Type Definition (OTD). The OTD is based on JAXP1.1, and any compliant XSLT processor and XML parser can be used with it. The preferred processor and parser are Xalan and Xerces, respectively, because of their close compliance with the XSLT and XML specifications, and these are the ones supplied with the product.

A set of OTD configuration parameters (see [XSLT Collaboration](#) on page 7) determine:

- The XSLT processor and XML parser to be used
- The style sheets that can be used by the Collaboration at run time
- The JDBC driver to be used for Cross-Reference Database access
- The user, password, and URL for accessing the database through JDBC

The OTD includes nodes and methods that facilitate the Collaborations—within the UAN framework—to perform XML message transformation, parsing, and validation, and cross-referencing of IDs and values.

### XML Message Transformation

The transform method performs the required transformation using the specified style sheet. It is available with different signatures:

- **transform()** transforms the content of the input buffer and places the result in the output buffer
- **transform(*source*)** transforms the parameter *source* as an XML message, and returns the result either as a byte array or a DOM document object, depending upon the form of *source*

### XML Message Parsing

The parse method performs the required parsing of the message. It is available with different signatures:

- **parse()** parses the content of the input buffer as an XML message and returns a DOM document object
- **parse(*xml*)** parses the message specified by the parameter *xml* and returns a DOM document object; the message may be either a string or a byte array

## XML Message Validation

The **validate** method validates the message in the input buffer, and returns a boolean true or false. It is available with different signatures:

- **validate()** assumes the project is referenced in the XML instance and is accessible by the XSLT processor
- **validate(*xsd*)** uses an external project as specified by the parameter *xsd*
- **validate(*xml*, *xsd*)** validates the message specified by the parameter *xml* using an external project as specified by the parameter *xsd*; *xml* can be expressed as either a string or a byte array

## ID Cross-Reference Mapping

The following methods, executed during the XSLT Collaboration service, return application information:

- GetAppID
- GetCommonID
- SetCommonID

## Value Cross-Reference Mapping

The following methods are associated with application routing and generation:

- GetAppValue
- GetCommonValue

### 1.1.3 Cross-Reference Database

ID and value cross-reference information is maintained in a relational database, consisting of a set of tables that contain the mapping information linking application-specific IDs and values and common-object IDs and values. These tables are created, and seeded with initial data, using SQL scripts. See the Siebel *Implementation and Configuration Guide: Universal Application Network Volume 2 (For SeeBeyond)*.



---

## 1.2 Supported Operating Systems

The Intelligent Adapter for Siebel UAN is available for the following operating systems:

- Windows XP, Windows 2000, or Windows Server 2003
- Solaris 8 and 9
- IBM AIX 5.1

In addition to the above listed Operating Systems, this eWay in outbound mode is supported on WebSphere™ Application Servers when using Java Collaborations only. For additional information, see the *eGate Integrator User's Guide*.

---

## 1.3 Host System Requirements

To use the Siebel UAN eWay, you need the following host system requirements.

### Logical Host Requirements

- eGate Integrator
- A TCP/IP network connection

**Note:** Before installing the eWay, open and review the *Readme.txt* file located on the Installation CD-ROM for any additional requirements.

---

## 1.4 External System Requirements

The Siebel UAN eWay requires an external RDBMS to support the cross-reference database.

Functional relational databases include:

- Oracle Database
- MS SQL Server
- DB2 Database

# Installing the Siebel UAN eWay

This chapter describes the installation procedure for the Siebel UAN eWay. For additional information, see the *eGate Integrator Installation Guide*.

### This Chapter Includes:

- [Overview](#) on page 10
- [Installing on Windows Operating Systems](#) on page 11

---

## 2.1 Overview

The installation process includes the following:

- Installing the Repository.
- Uploading products to the Repository.
- Downloading components (including Enterprise Designer and Logical Host).
- Viewing product information home pages.

Please review [Host System Requirements](#) on page 9 before installing the eWay.

---

## 2.2 Installing on Windows Operating Systems

**Note:** *You must have Administrator privileges to install this eWay.*

To install the Siebel UAN eWay on a Windows operating system:

- 1 Exit all Windows programs.
- 2 Insert the eGate installation CD-ROM into the CD-ROM drive.
- 3 Perform the following steps, following the installation instructions provided in the *ICAN Installation Guide*:
  - A Install and start the Repository.
  - B Upload **SiebelUANeWay.sar** to the Repository.
  - C Install the Enterprise Designer and Logical Host components.

# Setting Properties of Siebel UAN eWay

This chapter explains how to create and configure the Siebel UAN eWay properties.

---

## 3.1 Creating and Configuring the Siebel UAN eWay

The Siebel UAN eWay contains a unique set of default configuration parameters. After the eWays are established and a Siebel UAN External System is created in the Project's Environment, the configuration parameters are modified from two locations:

- From the **Connectivity Map**—which contains parameters specific to the Siebel UAN eWay.
- From the **Environment Explorer** tree—which contains global parameters that commonly apply to other eWays or eWays (of the same type) in the project.

---

## 3.2 Configuring the eWay Connectivity Map Properties

When you link an external application with a Collaboration, Enterprise Designer automatically assigns a template containing default configuration properties. For Siebel UAN, these properties are referred to as the Siebel UAN outbound eWay properties.

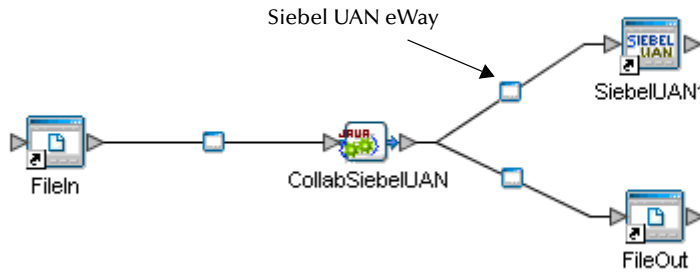
### 3.2.1 Siebel UAN Outbound eWay Properties

The Siebel UAN eWay contains only outbound parameters accessible via the Configuration Editor.

To configure the eWay properties:

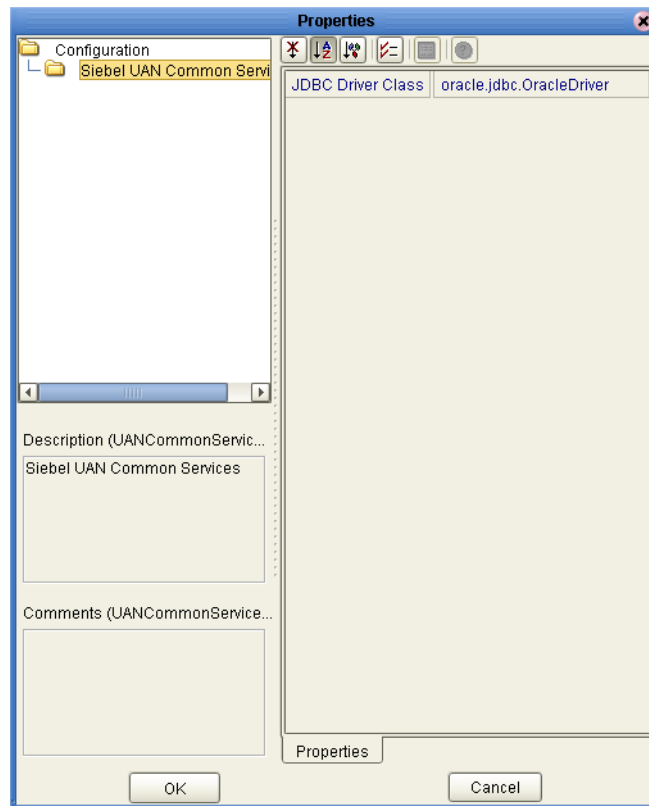
- 1 On the Enterprise Designer's Connectivity Map, double-click the inbound Siebel UAN icon. See [Figure 3 on page 13](#).

Figure 3 Connectivity Map



- 2 The configuration **Properties** window opens, displaying the default properties for the eWay.

Figure 4 Configuration Editor: Siebel UAN outbound eWay



- 3 Click on any folder to display the default configuration properties for that section.
- 4 Click on any property field to make it editable.
- 5 After modifying the configuration properties, click **OK** to save the changes.
- 6 Your modifications are now applied to the outbound eWay.

## 3.2.2 eWay Property Settings

The eWay property settings define how the properties are used to interact with the Siebel application. Configure your eWay property to match the property listed below.

### JDBC Driver Class

#### Description

This parameter specifies the vendor-specific JDBC Driver class.

#### Required Values

A fully-qualified class name; the default value is **oracle.jdbc.OracleDriver**.

---

## 3.3 Configuring the Environment Properties

The eWay Environment Configuration properties contain the parameters that define how the eWay connects to and interacts with other eGate components within the environment.

### 3.3.1 Siebel UAN External System Outbound Properties

The Siebel UAN environment only contains outbound properties that are accessible via the Environment Explorer.

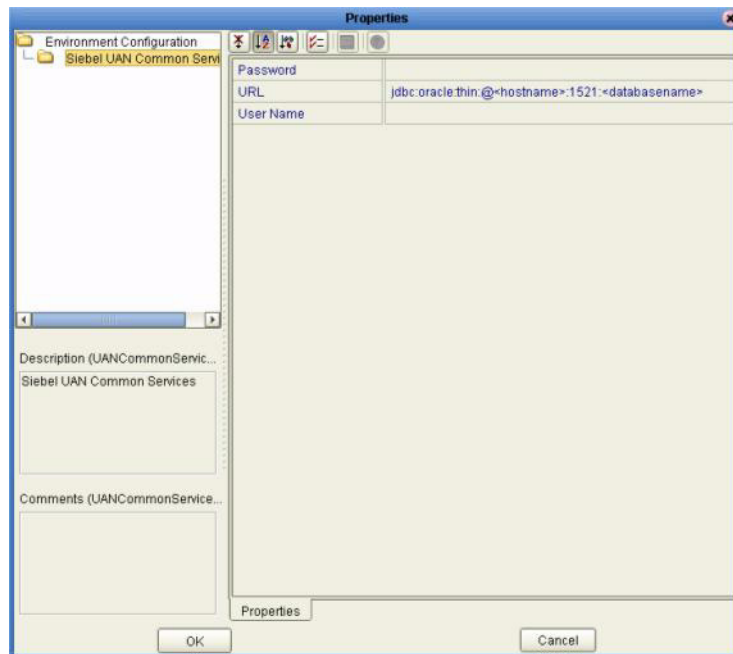
#### To configure the environment properties:

- 1 In Enterprise Explorer, click the **Environment Explorer** tab.
- 2 Expand the environment created for the Siebel UAN project and locate the **SiebelUAN External System**.

**Note:** For more information on creating an Environment, see the eGate Tutorial.

- 3 Right-click the SiebelUAN External System and select **Properties** from the list box. The Environment Configuration Properties window appears. See **Figure 5 on page 15**.

**Figure 5** Environment Configuration Outbound Properties



- 4 Click on any folder to display the default configuration properties for that section.
- 5 Click on any property field to make it editable.
- 6 After modifying the configuration properties, click **OK** to save the changes.

### 3.3.2 Environment Property Settings

Configure your environment properties to match the properties listed below.

#### URL

##### Description

This parameter specifies the JDBC URL for identifying the database where all persistent data for UAN Common Services are stored.

##### Required Values

Enter valid values for <hostname> and <databasename> as noted in the following URL.

jdbc:oracle:thin:@<hostname>:1521:<databasename>

#### User Name

##### Description

Specifies the user ID used to authenticate access to the database for UAN Common Services.

### **Required Values**

A valid user ID; there is no default value.

### **Password**

#### **Description**

This parameter specifies the password for authenticating access to the database for UAN Common Services.

### **Required Values**

A valid user password; there is no default value.



# Index

## A

Add-on properties 12

## C

configuration parameters

JDBC Driver Class 14

Password 16

URL 15

User 16

Configuring the Add-on 12

Connectivity Map 12

Connectivity Map Properties 12

Cross-Reference Database 8

## D

DB2 Database 9

## E

Environment Explorer 12

External System Requirements 9

## G

GetAppID 8

GetAppValue 8

GetCommonID 8

GetCommonValue 8

## H

Host System Requirements 9

## I

IBM AIX 5.1 9

Installing on Windows Operating Systems 11

## J

JDBC Driver Class 14

JDBC Driver Class parameter 14

## L

Logical Host Requirements 9

## M

MS SQL Server 9

## O

Operating Systems 9

Oracle Database 9

## P

parse (xml) 7

parse() 7

Password parameter 16

## S

SetCommonID 8

SiebelUANeWay.sar 11

Sun Solaris 8 9

Sun Solaris 9 9

Supported Operating Systems 9

System Requirements 9

## T

TCP/IP 9

transform() 7

transform(source) 7

## U

URL parameter 15

User parameter 16

## V

validate() 8

validate(xml, xsd) 8

validate(xsd) 8

Value Cross-Reference Mapping 8

## W

Windows 2000 9

Windows Server 2003 9

Windows XP 9

**X**

- XML Message Parsing 7
- XML Message Transformation 7
- XML Message Validation 8