SUN SEEBEYOND

# eVIEW™ STUDIO UPGRADE GUIDE

**Release 5.1.1**

Sun microsystems

Version 20060622155317

# Contents

Chapter 1

# Introduction

This chapter provides an overview of this guide and the conventions used throughout, as well as a list of supporting documents and information about using this guide.

**What's in this Chapter**



## 1.1 About the Upgrade Process

This guide provides the background information and instructions you need to upgrade the Sun SeeBeyond eView™ Studio (eView Studio) from version 5.0.5 or 5.1.0 to version 5.1.1. This process includes exporting your current eView Studio Projects from the previous environment, installing the latest version of eView Studio into the 5.1.1 Java Composite Application Platform Suite (CAPS) environment, and then importing the current Projects into the new Java CAPS environment.

If you are upgrading from version 5.1.0, perform the steps outlined in **Chapter 3 "Upgrading From Version 5.1.0"** after upgrading the software. If you are upgrading from version 5.0.5, perform the steps outlined in **Chapter 4 "Upgrading from Version 5.0.5"** after upgrading the software. Most of the steps are optional for the upgrade from 5.0.5, but must be performed in order to use new features provided after 5.0.5. If you are upgrading from a version earlier than 5.0.5, first upgrade to version 5.0.5 as described in the *Sun SeeBeyond eView Studio User's Guide*, version 5.0.5.

## 1.2 What's New in This Release

This release provides support for Oracle 10g, as well as improved performance for certain backend and concurrent transactions. For complete information about the changes included in this release, see the *eView Studio Release Notes*.

## 1.3 About This Document

This guide provides comprehensive information on upgrading eView Studio from version 5.0.5 or 5.1.0 to version 5.1.1. It includes instructions on installing the upgrade and on updating an eView Studio application by reconfiguring Project files.

The sections below provide information about this document, including an overview of its contents, scope, and intended audience.

### 1.3.1 What's in This Document

This guide is divided into the chapters that cover the topics shown below.

- **Chapter 1 "Introduction"** gives a general preview of this document—its purpose, scope, and organization—and provides sources of additional information.
- **Chapter 2 "Installing the Upgrade"** gives information and instructions for exporting and importing existing Projects, installing the latest eView Studio modules, and updating the Project files.
- **Chapter 3 "Upgrading From Version 5.1.0"** gives information and instructions for completing the upgrade from version 5.1.0.
- **Chapter 4 "Upgrading from Version 5.0.5"** gives information and instructions for completing the upgrade from version 5.0.5.

### 1.3.2 Scope

This guide provides step-by-step instructions for upgrading eView Studio. It includes navigational information, functional instructions, and background information where required. This guide does not include information or instructions on using the EDM or working with Enterprise Designer or eGate Integrator components. These topics are covered in the appropriate user guide (for more information, see **"Related Documents" on page 8**).

### 1.3.3 Intended Audience

Any user who upgrades any component of eView Studio should read this guide. A thorough knowledge of eView Studio is not needed to understand this guide. It is presumed that the reader of this guide is familiar with the eGate environment and GUIs, eGate Projects, Oracle database administration, and the operating system(s) on

which eGate and the eView Studio database run. Familiarity with XML documents, the SQL scripting language, and Java is helpful for the upgrade.

The intended reader must have a good working knowledge of his or her company's current business processes and information system (IS) setup, as well as the current eView Studio implementation.

### 1.3.4 Text Conventions

The following conventions are observed throughout this document.

**Table 1** Text Conventions

| Text Convention | Used For | Examples |
|---|---|---|
| **Bold** | Names of buttons, files, icons, parameters, variables, methods, menus, and objects | ▪ Click **OK**.<br>▪ On the **File** menu, click **Exit**.<br>▪ Select the **eGate.sar** file. |
| `Monospaced` | Command line arguments, code samples; variables are shown in ***bold italic*** | `java -jar` ***`filename`***`.jar` |
| **Blue bold** | Hypertext links within document | See **Text Conventions** on page 8 |
| Blue underlined | Hypertext links for Web addresses (URLs) or email addresses | http://www.sun.com |

### 1.3.5 Screenshots

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

## 1.4 Related Documents

Sun has developed a suite of user's guides and related publications that are distributed in an electronic library. The following documents might provide information useful in creating your customized index. In addition, complete documentation of the eView Studio Java API is provided in Javadoc format.

- *Sun SeeBeyond eView Studio User's Guide*
- *Sun SeeBeyond eView Studio Configuration Guide*
- *Sun SeeBeyond eView Studio Reference Guide*
- *Sun SeeBeyond Enterprise Data Manager User's Guide*
- *Implementing the Sun SeeBeyond Match Engine with eView Studio*
- *Sun SeeBeyond eView Studio Reporting Guide*

- *Sun SeeBeyond eView Studio Release Notes*
- *Sun SeeBeyond Java Composite Application Platform Suite Installation Guide*

## 1.5 Sun Microsystems, Inc. Web Site

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

http://www.sun.com

## 1.6 Documentation Feedback

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

CAPS_docsfeedback@sun.com

Chapter 2

# Installing the Upgrade

To begin the upgrade, you must upload the latest version of eView Studio to the Repository and install it to Enterprise Designer. You must first export your existing Projects from the old environment. After the software is upgraded, you can import the Projects into the updated 5.1.1 environment. This chapter provides instructions for performing these tasks.

**What's in this Chapter**



## 2.1 Supported Operating Systems

For this release, there were no changes to the supported operating systems. The eView Studio **Readme.txt** file (located in the **eViewDocs.sar** file) contains the most up-to-date information for the supported platforms.

## 2.2 System Requirements

eView Studio 5.1.1 supports different integration software platforms than version 5.1.0. If you need to upgrade any of the following components, be sure to complete the upgrades before beginning the eView Studio upgrade. To upgrade the required Java CAPS components, see the *Sun SeeBeyond Java Composite Application Platform Suite Installation Guide* and the user's guide for any eWays to be upgraded.

The eView Studio **Readme.txt** file (located in the **eViewDocs.sar** file) contains the most up-to-date information for system requirements.

### 2.2.1 Java Composite Application Platform Suite Requirements

eView Studio 5.1.1 must be installed with eGate Integrator version 5.1.1. The File eWay version 5.1.1 and Oracle eWay version 5.1.1 are also required (the Oracle eWay is required for database connectivity when running on the Sun SeeBeyond Integration

Server (IS); it is optional when running on the Sun Java System Application server). eView Studio supports both Oracle Thin and Oracle OCI drivers. See the *Sun SeeBeyond eWay Adapter for Oracle User's Guide* for more information about working with Oracle drivers.

Your current eView Studio environment must be at version 5.0.5 or later in order to perform the upgrade. If your environment is currently at a previous version, see chapter 3 of the *Sun SeeBeyond eView Studio User's Guide*, version 5.0.5. Before performing the eView Studio upgrade, make sure to upgrade the Java CAPS environment as described in the *Sun SeeBeyond Java Composite Application Platform Suite Installation Guide.*

### 2.2.2 Database Software Requirements

Previous versions of eView Studio supported Oracle 8*i* and 9*i*. This version of eView Studio supports Oracle 9*i* and 10*g*. If you need to upgrade Oracle, do so prior to performing the eView Studio upgrade.

### 2.2.3 Application Server Requirements

eView Studio 5.1.1 can be deployed on the Sun Java System Application Server 8.1 or the Sun SeeBeyond Integration Server 5.1.1.

## 2.3 Upgrading the eView Studio Environment

Before beginning, make sure your Java CAPS environment, including eGate Integrator, eWays, libraries, and so on, have all been upgraded to 5.1.1. If you did not previously have the Oracle eWay installed and will be using it for database connectivity, install the eWay now.

Perform the following steps to upgrade the eView Studio modules and Projects.

- **Step 1: Upgrade Oracle** on page 11
- **Step 2: Export Existing Projects and Environments** on page 12
- **Step 3: Install eView Studio 5.1.1** on page 12
- **Step 4: Import the Exported Projects and Environments** on page 12

Once you complete the above steps you can manually update certain components of the eView Studio Project to use the new features. If you are upgrading from version 5.1.0, this is described in **Chapter 3 "Upgrading From Version 5.1.0"**. If you are upgrading from version 5.0.5, this is described in **Chapter 4 "Upgrading from Version 5.0.5"**.

### 2.3.1 Step 1: Upgrade Oracle

If you are currently running Oracle 8*i*, upgrade to version 9*i* or 10*g* before performing any of the following steps. Similarly, if you are currently running Oracle 9*i* and want to

upgrade to 10*g*, do so before continuing. For more information about upgrading Oracle, see the Oracle documentation.

## 2.3.2 Step 2: Export Existing Projects and Environments

From your 5.0.5 or 5.1.0 Enterprise Designer, export any existing eView Studio server and client Projects that you want to upgrade. You can either export your existing Environments or create new ones after you upgrade eView Studio.

This section describes how to export the Projects using Enterprise Designer; you can also export the Projects from a command line (see the *Sun SeeBeyond eGate Integrator User's Guide* for more information).

**To export a Project or Environment using Enterprise Designer**

1. From the Repository context menu, select **Export Project** to display the Export Manager dialog.
2. Highlight the desired Project(s) and Environment(s) in the displayed list and transfer them to the *Selected Projects* and *Selected Environments* panel using the arrow buttons.
3. Click the **Browse** button to display the *Save As* dialog.
4. Select the export destination and change the export file name, if desired.
5. Click **Save As** to accept the destination path and file name.
6. Click **Export** to export the Project file (this process might take a few minutes).

   The Export Status message box appears after the file is exported successfully.
7. Click **OK** to close the message box.

## 2.3.3 Step 3: Install eView Studio 5.1.1

Before importing the exported Project files into the 5.1.1 environment, install eView Studio 5.1.1 in your 5.1.1 Repository and Enterprise Designer. Follow the instructions provided in chapter 3 of the *Sun SeeBeyond eView Studio User's Guide* to install eView Studio. That chapter also provides information for installing the reports, the updated eView Studio sample, documentation, and Javadocs.

## 2.3.4 Step 4: Import the Exported Projects and Environments

This section describes how to import the Projects and Environment you previously exported using Enterprise Designer. You can also export the Projects from a command line (see the *Sun SeeBeyond eGate Integrator User's Guide* for more information).

**To import a Project using Enterprise Designer**

1. From the Repository context menu, select **Import Project**.
2. A message box appears, prompting you to save any unsaved changes to the Repository.

A If you want to save your changes and have not already done so, click **No**. Save your changes, and then re-select **Import Project**, as in step 1.

B If you have saved all changes, click **Yes**.

3 Click the **Browse** button to display the Open File dialog. If you browse to an Environment file, the **Root environment** field is enabled.

4 Locate and select the Project or Environment file that you want to import.

5 Click **Open** to import the file.

The Import Manager dialog appears.

***Note:*** *If the Project you are importing contains references to another Project and the other Project already resides in your Repository, you have the option of excluding the referenced Project from the import by checking the box that appears in the Exclude column. The references will be retargeted to the Project existing in the 5.1.1 Repository.*

6 Click **Import** to import the file.

The Import Status message box appears after the file is imported successfully.

7 Click **OK** to close the message box.

8 When you are finished importing files, click **Close** to close the Import Manager dialog. The Project Explorer is automatically refreshed from the Repository.

9 Do one of the following:

- If you are upgrading from version 5.1.0, update the Project files as described in Chapter 3 **"Upgrading From Version 5.1.0"**.
- If you are upgrading from version 5.0.5, update individual Project components as described in Chapter 4 **"Upgrading from Version 5.0.5"**.

Chapter 3

# Upgrading From Version 5.1.0

Once you import the eView Studio Projects and Environments, you can update the components to take advantage of the new functions provided in this release. This chapter provides general information about the types of updates to perform. Any applications created after you upgrade eView Studio will automatically contain the updates.

**What's in this Chapter**



## 3.1 Updating the Project Files

For this release, the United States first name configuration file for the standardization engine was improved with new names and nicknames. Several existing names were remapped to better reflect actual usage.

To use the new configuration files for the standardization engine in existing Projects, you must load them into each eView Studio server Project to bring them into the Repository. You only need to perform this step if you want to use the updated United States first name standardization file, which is stored in the **United States** folder of the **Standardization Engine** node. After you upgrade eView Studio, the new file is automatically loaded into any new Projects you create.

Be aware that the differences between the 5.1.0 and the 5.1.1 versions are extensive, with many new entries, changes to existing entries, and deletions. You can use a standard diff tool to view the differences between the two versions. To do this, export the file (**personFirstNameUS.dat**) from the Project prior to loading the new file, load and then export the new file, and then compare the two files.

***Important:*** *Changes to the configuration files change how data is standardized, which can result in data that was standardized prior to using the updated files not being picked up by the blocking query after the update. In many cases the phonetic codes are still the same and would still be picked up by the query. You can write a SQL script to update any normalized or phonetically encoded data that was standardized prior to*

*updating the configuration files. Contact Sun SeeBeyond for help in writing the SQL script.*

**To load standardization files**

1. In the Project Explorer in Enterprise Designer, expand the eView Studio Project, and then expand the eView Studio application.
2. Right-click the **Standardization Engine** folder, and then select **Load Configuration Files** from the context menu.
3. In the **Open** dialog, open the folder containing the files you want to load.
4. Navigate to the United States subdirectory and then select **personFirstNameUS.cfg**.
5. Click **Open**.
6. On the Information dialog, click **OK**.

# 3.2 Working With the Updated Java API

Java methods were updated with this release of eView Studio to improve transaction handling and to give you more flexibility. If you are using any of these methods in your Java Collaborations or external clients, you might want to update them with the new methods.

For more information about any of the new methods or classes, refer to the Javadocs provided with eView Studio.

## updateSystemObject

This release adds one new **updateSystemObject** method to the MasterController class to better handle instances where a record is being updated at the same time by two different users or processes. The syntax of the new method is:

```
updateSystemObject(SystemObject sysobj, String revisionNumber)
```

## AssumedMatchIterator and PotentialDuplicateIterator Classes

Two new methods, **setReadForwardOnly** and **sortSummaryBy**, were added to the AssumedMatchIterator and PotentialDuplicateIterator classes. **setReadForwardOnly** specifies whether the iterator will clear the objects of a loaded page once it starts to read and load the next page. **sortSummaryBy** sorts the objects in the iterator by the specified comparator without loading the associated objects. The syntax of each new method is:

```
setReadForwardOnly(boolean forwardOnly)
```

and

```
void sortSummaryBy(String field, boolean reverse)
```

## 3.3 Completing the Project Upgrade

To ensure any changes you made take effect and to update the .jar files to version 5.1.1, perform the following tasks in the order given.

- **Regenerating the Application** on page 16
- **Updating the Collaborations** on page 16
- **Recreating the Deployment Profiles** on page 16
- **Redeploying the Projects** on page 17

### 3.3.1 Regenerating the Application

Before deploying the eView Studio application to the new Environment, regenerate the application to incorporate any changes you made. To regenerate, right click on the eView Studio application and then select **Generate**. Do this for each eView Studio application that was upgraded.

### 3.3.2 Updating the Collaborations

For each Collaboration client Project that references the eView Studio Java API, you must refresh the Java Collaborations to use the updated .jar files from the regenerated eView Studio server Project.

**To update the Collaborations**

1. In the Project Explorer, check out and open the Collaboration to be updated.
2. In the toolbar, click **Import JAR File**. The Add/Remove Jar Files dialog appears.
3. For each eView Studio .jar file in the list, highlight the filename and then click **Remove**.
4. For each eView Studio .jar file to re-import, do the following
   - A Click **Add**.
   - B Double-click the **eView_Sample_Collab_Client** Project name in the list that appears.
   - C Select the name of the file to import.
   - D Click **Import**.

### 3.3.3 Recreating the Deployment Profiles

You must create a new Deployment Profile for each upgraded Project, and then build and deploy each Project. Deployment Profiles for eView Studio Projects are described in the *Sun SeeBeyond eView Studio User's Guide*. If you did not import the Environment, you must recreate the Environment for the eView Studio Projects before building or deploying.

Once all Project components are mapped to the environment, build the Projects. For information about building a Project, see the *Sun SeeBeyond eView Studio User's Guide*.

### 3.3.4 Redeploying the Projects

Once the eView Studio Projects are rebuilt, redeploy the Projects to the server. You can use either Enterprise Designer or Enterprise Manager to perform this task. For information and instructions on redeploying Projects, see the *Sun SeeBeyond eView Studio User's Guide*. For information about deploying from Enterprise Manager, see the *Sun SeeBeyond eGate Integrator System Administration Guide*.

## 3.4 Updating Command Line Reports

If you use the command line reports, Sun recommends reinstalling the reports. See "Downloading eView Studio Reports" in the *Sun SeeBeyond eView Studio User's Guide* for instructions on installing the updated reports.

After you install the reports and regenerate and deploy the eView Studio server Project, export the eView Studio server Project files **<project_name>_stc_eindex_client.jar** and **<project_name>_stc_eindex_util.jar** to the **\lib** directory of the reports home directory. Rename the files to **stc_eindex_client.jar** and **stc_eindex_util.jar**.

Chapter 4

# Upgrading from Version 5.0.5

Once you import the eView Studio Projects and Environments, you can update the components to take advantage of the new functions provided in release 5.1.0. This chapter provides general information about the types of updates to perform. Any applications created after you upgrade eView Studio will automatically contain the updates.

**What's in this Chapter**



***Important:*** *Many of the changes described in the following sections were also implemented in eView Studio 5.0.5 Emergency Software Releases (ESRs). This document describes how to implement all changes from the baseline version of eView Studio 5.0.5 with no ESRs installed.*

## 4.1 Updating the Project Files

For this release, several changes were made to the eView Studio Project configuration files. All of the changes are optional, but must be implemented to use the new functionality of this release. The following configuration files were modified: the Enterprise Data Manager file, the Match Field file, the Threshold file, and the Candidate Select file. In addition, the United Kingdom first name configuration file for the standardization engine was improved and a new parameter was added to the business constants file.

## 4.1.1 Enterprise Data Manager File

This section describes how to update the Enterprise Data Manager file for each of the Enterprise Data Manager (EDM) enhancements. Perform any of the following procedures to update the Enterprise Data Manager file with the latest features.


- **Implementing EDM Reports** on page 19
- **Defining the Initial View** on page 21
- **Configuring System Record Merges** on page 22
- **Defining Range Searching for the EDM** on page 23
- **Defining Required Fields for a Search** on page 23


All of these changes are optional, but must be implemented in order to use the new functions. For more information about the new functions, see chapter 9 of the *Sun SeeBeyond eView Studio Configuration Guide*.

### Implementing EDM Reports

This release supports running standard reports from the EDM as well as from a command line. The EDM Reports page and each report are configured in the Enterprise Data Manager file.

#### To configure the EDM Reports page

For the Reports page, define the parent object, the name of the tabbed heading for the page, the URL to access the Reports page, and the number of fields on each row of the Reports Search page. All of the elements mentioned below are described in Table 2.

1. In the Enterprise Data Manager file, scroll to the **matching-review** element (located near the end of the file) and add new **reports** tag underneath, but at the same level as, the closing **matching-review** tag, as shown below.

```
<matching-review>
<root-object>Company</root-object>
   <tab-name>Matching Review</tab-name>
   <tab-entrance>/EnterPDSearchAction.do</tab-entrance>
   <pd-search-page>
      <field-per-row>2</field-per-row>
   </pd-search-page>
   <search-result-list-page>
      <item-per-page>10</item-per-page>
      <max-result-size>100</max-result-size>
   </search-result-list-page>
</matching-review>
<reports>
</reports>
```

2. Within the **reports** tags, create and define the elements defined in Table 2, as shown in the example below.

```
<reports>
   <root-object>Company</root-object>
   <tab-name>Reports</tab-name>
   <tab-entrance>/EnterReportSearchAction.do</tab-entrance>
   <search-page-field-per-row>2</search-page-field-per-row>
</reports>
```

3 After you finish configuring the Reports page, save the file.

**Table 2** Reports Page Configuration Elements

| Element | Description |
|---|---|
| root-object | The name of the type of object on which to report (this must be the parent object). |
| tab-name | A name for the report pages. This name appears on tab label associated with the report pages on the EDM, and is typically "Reports". |
| tab-entrance | The URL to the entry page of the reports pages. This element must be defined as "/EnterReportSearchAction.do". |
| search-page-field-per-row | The number of fields to display in each row of the Reports Search page. |

### To configure the EDM reports

You must define a configuration section for each report you want to run from the EDM. There are six production reports and three activity reports to define. The elements and attributes mentioned in the following instructions are described in **Table 3 on page 21**.

Perform the following steps for each production and activity report. Complete the steps outlined under **"To configure the EDM Reports page" on page 19** first.

1 In the Enterprise Data Manager file, scroll to the newly created **reports** element.

2 Under the **search-page-field-per-row** element, define the elements and attributes described in Table 3. An example of one production and one activity report is shown below.

```
<reports>
   <root-object>Company</root-object>
   <tab-name>Reports</tab-name>
   <tab-entrance>/EnterReportSearchAction.do</tab-entrance>
   <search-page-field-per-row>2</search-page-field-per-row>
   <report name="Potential Duplicate"
    title="Potential Duplicates">
      <enable>true</enable>
      <max-result-size>1000</max-result-size>
      <page-size>500</page-size>
      <fields>
         <field-ref>Company.Name</field-ref>
         <field-ref>Company.Symbol</field-ref>
         <field-ref>Company.Exchange</field-ref>
         <field-ref>Company.SIC</field-ref>
         <field-ref>Address.AddressLine1</field-ref>
         <field-ref>Phone.Phone</field-ref>
      </fields>
   </report>
     ...
   <report name="Weekly Activity"
    title="Weekly Transaction Report">
      <enable>true</enable>
      <max-result-size>2000</max-result-size>
      <fields></fields>
   </report>
</reports>
```

3 After you configure each report, save, validate, and close the file.

**Table 3** Report Configuration Elements

| Element/Attribute | Description |
|---|---|
| report | Defines each report run by the EDM with the exception of search reports (which do not need to be configured). This element is defined by two attributes: name and title. |
| report/name | The type of report being generated. Specify any of the following production reports.<br>▪ Assumed Match<br>▪ Potential Duplicate<br>▪ Deactivated<br>▪ Merged<br>▪ Unmerged<br>▪ Update<br>Or specify any of the following activity reports.<br>▪ Weekly Activity<br>▪ Monthly Activity<br>▪ Yearly Activity |
| report/title | The descriptive name of the report. This can be any string, and appears as the title in the specified report. |
| enable | Specifies whether the report can be run from the EDM. Specify "true" to allow the report to be run; specify "false" to disable the report from the EDM. |
| max-result-size | The number of records to display on the report. If no value is entered, or if the value is zero (0), the size defaults to 1000 records. To make sure all records are retrieved for a report, enter a very large value for this element. |
| page-size | The number of records returned to the report generator at one time for each report. If you do not enter a page size or you enter "0", the size defaults to 500 records for all reports. |
| fields/field-ref | A list of fields to display on the report in addition to those that are displayed automatically. Use the simple field name for the **field-ref** value (simple field names are described in appendix B of the *Sun SeeBeyond eView Studio Configuration Guide*). This element should be empty for the activity reports; if a list of fields is supplied for any activity report, it is ignored. |

## Defining the Initial View

You can specify any of the tabbed pages on the EDM as the first page to appear after a user logs on. If you do not add this section to the Enterprise Data Manager file, the default is the Matching Review page.

### To define the initial view

1 In the Enterprise Data Manager file, scroll to the **page-definition** element in the **gui-definition** element.

2. Add a new element, **initial-screen**, immediately following the **page-definition** element (at the same level as the **eo-search** element).
3. Change the value of the **initial-screen** element to one of the following:
   - Matching Review - to display the Matching Review Search page first.
   - EO Search - to display the Search page first.
   - Create System Record - to display the Create System Record page first.
   - History - to display the History/Audit Log Search page first.
   - Reports - to display the Reports Search page first.

***Note:*** *These values remain the same regardless of whether the tab name of the page has been changed. For example, if you change the name of the Matching Review page to "Potential Duplicate", you would still specify "Matching Review" for that page to appear first.*

4. Add a closing **initial-screen** tag. A sample appears below.

```
<gui-definition>
   ...
   <page-definition>
      <initial-screen>Matching Review</initial-screen>
      <eo-search>
```

5. Save and close the file.

## Configuring System Record Merges

This release provides the ability to specify whether an EDM user must select the child objects to retain during a system record merge, or whether the user can accept the default child objects. (The default child objects are those contained in the destination system record.)

### To configure system record merges

1. In the Enterprise Data Manager file, scroll to the **node-<object>** element that defines the child object you want to configure.
2. Add a new attribute, **merge-must-delete**, to the **node-<object>** element and specify "true" or "false" for the attribute (see sample below).

```
<node-Phone display-order="3" merge-must-delete="true">
   <field-PhoneType>
      <display-name>Phone Type</display-name>
      <display-order>1</display-order>
      <max-length>8</max-length>
      <gui-type>MenuList</gui-type>
      <value-list>PHONTYPE</value-list>
      <value-type>string</value-type>
      <key-type>true</key-type>
   </field-PhoneType>
   ...
</node-Phone>
```

***Note:*** *Set this element to "true" for a child object type to force manual selection of the child objects to retain for that object type. Set it to "false" to give the option of either*

*accepting the default objects for the merge or selecting them manually. This attribute is optional and defaults to "false" if it does not exist for a child object or if no value is defined.*

3 Save and close the file.

## Defining Range Searching for the EDM

In the Enterprise Data Manager file, you can specify that certain fields can be used to search by a range of values, where the user supplies the upper and lower values of the range. For blocking queries, additional configuration is required in the Candidate Select file (see **"Defining Range Searching for Blocking Queries" on page 30**).

### To define range searching

1 In the Enterprise Data Manager file, scroll to the **page-definition** element in the **gui-definition** element.

2 Scroll to the **eo-search** element, and then to the **simple-search-page** element you want to define for range searching.

3 In the **simple-search-page** element, scroll to the **field-ref** you want to define for range searching, and then add a new attribute named "choice".

4 Set the **choice** attribute to "range", as shown below.

```
<field-group>
   <description>Company</description>
   <field-ref>Company.Name</field-ref>
   <field-ref>Company.Symbol</field-ref>
   <field-ref>Company.Exchange</field-ref>
   <field-ref>Company.SIC</field-ref>
   <field-ref choice="range">Company.ContactDate</field-ref>
</field-group>
```

5 Save, validate, and close the file.

## Defining Required Fields for a Search

In the Enterprise Data Manager file, you can configure a specific field to be required or you can configure a group of fields of which at least one is required. Required fields are defined individually for each search page, so a field that is required for one type of search might not be required for another.

### To define required search fields

1 In the Enterprise Data Manager file, scroll to the **page-definition** element in the **gui-definition** element.

2 Scroll to the **eo-search** element, and then to the **simple-search-page** element for which you want to define required search fields.

3 In the **simple-search-page** element, scroll to the **field-ref** element defining the field that should be required and add a new attribute named "required".

4 Set the **required** attribute to one of the values listed in Table 4, as shown below.

```
<field-group>
   <description>Company</description>
```

```
        <field-ref required="true">Company.Name</field-ref>
        <field-ref>Company.Contact</field-ref>
        <field-ref required="oneof">Company.Symbol</field-ref>
        <field-ref required="oneof">Company.Exchange</field-ref>
        <field-ref required="oneof">Company.SIC</field-ref>
    </field-group>
```

5 Save, validate, and close the file.

**Table 4** Values for the Required Attribute

| Attribute Value | Description |
|---|---|
| true | Indicates the field is required on the search page you are configuring. |
| false | Indicates the field is not required on the search page you are configuring (this is the default value). |
| oneof | Indicates at least one of the fields specified as “oneof” is required on the search page you are configuring. |

## 4.1.2 Threshold File

This section describes how to update the Threshold file to take advantage of the new ability to customize the logic of the execute match methods. To incorporate the custom plug-ins that define the customized logic, you must specify the name of the plug-ins in the Threshold file. For more information about the custom plug-ins required for the custom logic, see **Custom Plug-ins for executeMatch** on page 34 and **Customizing executeMatch** on page 39.

**To specify a custom plug-in for Collaborations and Business Processes**

1 In the Threshold file, scroll to the **MasterControllerConfig** element.

2 Create a new **logic-class** element, and specify the name of the custom plug-in that contains the back-end logic. For example:

```
<logic-class>com.stc.eindex.user.CustomProcessing</logic-class>
```

3 Save and close the file.

**To specify custom plug-in for the Enterprise Data Manager**

1 In the Threshold file, scroll to the **MasterControllerConfig** element.

2 Create a new **logic-class-gui** element, and specify the name of the custom plug-in that contains the EDM logic. For example:

```
<logic-class-gui>com.stc.eindex.user.CustomEDM</logic-class-gui>
```

3 Save and close the file

## 4.1.3 Match Field File

This section describes how to update the Match Field file to use the new French or Australian national domains or to implement standardization on multiple national domains. It also describes how to implement the new phonetic encoders and how to

modify address standardization for improved matching and queries. You can modify the Match Field file in the following ways.

- **Specifying a New National Domain** on page 25
- **Specifying Multiple National Domains** on page 25
- **Specifying new Phonetic Encoders** on page 27
- **Remapping Street Address Standardization** on page 28

All of these changes are optional, but must be implemented in order to use the new functions. For more information about the new functions, see chapter 6 of the *Sun SeeBeyond eView Studio Configuration Guide*.

## Specifying a New National Domain

You can modify your normalization and standardization structures to use one of the new national domains (France or Australia), however this might result in matching discrepancies between records that were processed prior to the change and records that were processed after.

### To specify a new national domain

1. In the Match Field file, scroll to the normalization or standardization structure you want to modify.
2. In the **group** element containing the structure you want to change, change the value of the **domain-selector** attribute to one of the following:
   - com.stc.eindex.matching.impl.SingleDomainSelectorAU - to use the Australian national domain.
   - com.stc.eindex.matching.impl.SingleDomainSelectorFR - to use the French national domain.

   A sample is shown below.

```
<structures-to-normalize>
   <group standardization-type="BusinessName"domain-selector=
    "com.stc.eindex.matching.impl.SingleDomainSelectorAU">
      <unnormalized-source-fields>
        ...
   </group>
</structures-to-normalize>
```

3. Save, validate, and close the file.

## Specifying Multiple National Domains

You can modify your normalization and standardization structures to use multiple national domains to standardize data, however this might result in matching discrepancies between records that were processed prior to the change and records that were processed after.

### To specify multiple national domains

1. In the Match Field file, scroll to the normalization or standardization structure you want to modify.

2 In the **group** element containing the structure you want to define to use multiple national domains, change the value of the **domain-selector** attribute to "com.stc.eindex.matching.impl.MultipleDomainSelector".

```
<structures-to-normalize>
   <group standardization-type="PersonName"domain-selector=
    "com.stc.eindex.matching.impl.MultipleDomainSelector">
      <unnormalized-source-fields>
       ...
   </group>
</structures-to-normalize>
```

3 In the **group** element, create a **locale-field-name** element and a **locale-maps** element.

4 For each national domain you want to use, define a **locale-codes**, **value**, and **locale** element (described in **Table 5 on page 27**). Following is an example of a multiple domain configuration.

```
<structures-to-normalize>
   <group standardization-type="PersonName"domain-selector=
    "com.stc.eindex.matching.impl.MultipleDomainSelector">
      <locale-field-name>Person.PobCountry</locale-field-name>
      <locale-maps>
         <locale-codes>
            <value>GB</value>
            <locale>UK</locale>
         </locale-codes>
         <locale-codes>
            <value>UNST</value>
            <locale>US</locale>
         </locale-codes>
         <locale-codes>
            <value>AU</value>
            <locale>AU</locale>
         </locale-codes>
         <locale-codes>
            <value>FR</value>
            <locale>FR</locale>
         </locale-codes>
            <value>Default</value>
            <locale>AU</locale>
         </locale-codes>
      </locale-maps>
      <unnormalized-source-fields>
      ...
```

5 Save, validate, and close the file.

**Table 5** Domain Configuration Elements

| Element | Description |
|---|---|
| locale-field-name | The ePath to an identifying field in the object structure that will identify which of the defined **local-codes** definitions to use. If no field is specified, the standardization engine defaults to the United States domain, regardless of whether any of the following elements are defined. This field must belong to the object that contains the fields defined for standardization in this structure. |
| **locale-maps elements** | |
| locale-codes | Each locale codes stanza defines a value (**value**) that can be contained in the identifying field (**locale-field-name**) in a transaction, and also defines that value's corresponding domain (**locale**). |
| value | A value that, when contained in the identifying field, indicates that the standardization engine will use the corresponding **locale** element to determine which national domain to use to standardize the data. You can specify a default domain by entering "Default" in the **value** element and one of the locale codes described below in the **locale** element. |
| locale | A domain code indicating which national domain to use to standardize data when the identifying field value in a transaction matches the corresponding **value** element. Use any of the following locale codes:<br>• **AU** - for Australian data<br>• **FR** - for French data<br>• **UK** - for United Kingdom data<br>• **US** - for United States data |

## Specifying new Phonetic Encoders

You can use any of the four new phonetic encoders to generate the phonetic codes used in searching and matching. If you change the encoder for a specific field, you should create a script that will update the phonetic versions of the data that already exists in your eView Studio database.

### To add the new phonetic encoders to the Matching Service

1. In the Match Field file, scroll to the **PhoneticEncodersConfig** section.
2. Create a new **encoder** element, and then define the elements described in Table 6, as shown below. The new implementation classes are listed in Table 7.

```
<encoder>
   <encoding-type>NYSIIS</encoding-type>
   <encoder-implementation-class>
    com.stc.eindex.phonetic.impl.Nysiis
   </encoder-implementation-class>
</encoder>
```

```
<encoder>
   <encoding-type>Metaphone</encoding-type>
   <encoder-implementation-class>
    com.stc.eindex.phonetic.impl.Metaphone
   </encoder-implementation-class>
</encoder>
```

3 Save, validate, and close the file.

**Table 6** PhoneticEncodersConfig Elements

| Element | Description |
|---|---|
| encoding-type | The name of the phonetic encoder, such as NYSIIS or Soundex. |
| encoder-implementation-class | The fully qualified name of the Java class that determines which phonetic encoder to use. |

**Table 7** Phonetic Encoder Classes for the SeeBeyond Match Engine

| Encoder | Java Class |
|---|---|
| Soundex | com.stc.eindex.phonetic.impl.Soundex |
| NYSIIS | com.stc.eindex.phonetic.impl.NYSIIS |
| Metaphone | com.stc.eindex.phonetic.impl.Metaphone |
| Double Metaphone | com.stc.eindex.phonetic.impl.DoubleMetaphone |
| Refined Soundex | com.stc.eindex.phonetic.impl.RefinedSoundex |
| French Soundex | com.stc.eindex.phonetic.impl.SoundexFR |

**To specify a new phonetic encoder for a field**

1 In the Match Field file, scroll to the **phoneticize-fields** element in the **PhoneticEncodersConfig** element.

2 For the field you want to modify, change the value of the **encoding-type** element to the name of the encoder you want to use. (The name of the encoder is defined in the **encoding-type** element in the **PhoneticEncodersConfig** section.)

3 Save, validate, and close the file.

## Remapping Street Address Standardization

In the default configuration generated by the eView Wizard in previous versions, the Match Field file defined the original version of the street name to be stored in the database instead of the standardized version. This can cause errors in searches and matching if you are using the **StreetName** field for matching or for queries. The correct street name is used in any Projects generated in 5.1.0 or later.

***Note:*** *To maintain the integrity of existing data, it is recommended that this only be performed on applications that have not yet gone into production.*

**To remap street address standardization**

1. In the Match Field file, find the following stanza in the address standardization structure, and change "OrigStreetName" to "MatchStreetName".

```
<target-mapping>
   <standardized-object-field-id>OrigStreetName
   </standardized-object-field-id>
   <standardized-target-field-name>Person.Address[*].StreetName
   </standardized-target-field-name>
<target-mapping>
```

2. Save, validate, and close the file.

## 4.1.4 Candidate Select File

This section describes how to update the Candidate Select file to use Oracle hints, range searching, and wildcard characters. It includes the following procedures:

- **Defining Oracle Hints** on page 29
- **Defining Range Searching for Blocking Queries** on page 30
- **Using Wildcard Characters in a Blocking Query** on page 31

All of these changes are optional, but must be implemented in order to use the new functions. For more information about the new functions, see chapter 4 of the *Sun SeeBeyond eView Studio Configuration Guide*.

### Defining Oracle Hints

You can add Oracle hints to blocking queries to help optimize query execution. Among other uses, hints are especially useful when a blocking query uses only child object fields; the hint can specify to scan the child object table first.

**To define an Oracle hint**

1. In the Candidate Select file in the Project you want to modify, scroll to the **query-builder** element that contains the blocking query to which you want to add a hint.
2. Create and define a new **hint** element in the **block-definition** element containing the query block to which the hint will apply.

```
<config>
   <block-definition number="ID1">
      <hint>FIRST_ROWS_100</hint>
      <block-rule>
         <equals>
            <field>Enterprise.SystemSBR.Company.Name</field>
            <source>Company.Name</source>
         </equals>
         <equals>
            <field>Enterprise.SystemSBR.Company.Symbol</field>
            <source>Company.Symbol</source>
         </equals>
      </block-rule>
   </block-definition>
<config>
```

3. Save, validate, and close the Candidate Select file.

## Defining Range Searching for Blocking Queries

Range searching is defined in both the Candidate Select file and the Enterprise Data Manager file. For basic queries, range searching is defined solely in the Enterprise Data Manager file (see **"Defining Range Searching for the EDM" on page 23**) and can only be simple range searches where both the upper and lower limits are provided by the user or an incoming message. Blocking queries support more complex range searches that can include offsets and constants, as well as user-specified upper and lower limits. Additional steps are required to configure range searching for blocking queries.

### To define range searching for a blocking query

1. In the Candidate Select file, scroll to the **query-builder** element that contains the blocking query to modify.
2. In each **block-definition** element containing a field to be defined for range searching, change the element named "equals" to "range" for that field only.
3. If using offsets or constants, define the **default** elements under the **range** element, as shown below. These elements are described in Table 8.

```
<config>
   <block-definition number="ID1">
      <hint>FIRST_ROWS_100</hint>
      <block-rule>
         <equals>
           <field>Enterprise.SystemSBR.Company.Name</field>
           <source>Company.Name</source>
         </equals>
         <range>
           <field>Enterprise.SystemSBR.Company.ContactDate</field>
           <source>Company.ContactDate</source>
           <default>
              <lower-bound type="offset">-5</lower-bound>
              <upper-bound type="constant">2007-01-01
              </upper-bound>
           </default>
         </range>
      </block-rule>
   </block-definition>
<config>
```

***Note:*** *To define a simple range search, where the user or incoming message supplies the upper and lower limits of the range, do not define any* ***default*** *elements in the* ***range*** *element; be sure to define that field for range searching in the Enterprise Data Manager file for any searches that use the query. If you are using offsets or constants only, do not define the field for range searching in the Enterprise Data Manager.*

4. Save and close the file.

**Table 8** Range Searching Configuration Elements

| Element/Attribute | Description |
|---|---|
| lower-bound/type | Defines the lower limit of a range search. Define the type attribute as “offset” to use an offset value, or as “constant” to define a lower constant. The value of the type attribute can be numeric, date, or string. See appendix A of the *Sun SeeBeyond eView Studio Configuration Guide* for more information. |
| upper-bound/type | Defines the upper limit of a range search. Define the type attribute as “offset” to use an offset value, or as “constant” to define an upper constant. The value of the type attribute can be numeric, date, or string. |

## Using Wildcard Characters in a Blocking Query

You can use wildcard characters in the blocking queries defined in the Candidate Select file. This is especially useful when fields from child objects are included in a blocking query because it allows the query to use the field value from each instance of the child object in an incoming message as criteria. If you do not use wildcard characters for fields in a child object and an incoming message contains more than one instance of that child object, the query arbitrarily chooses the child object to use as criteria.

**To use wildcard characters in a blocking query**

1. In the Candidate Select file in the Project you want to modify, scroll to the **query-builder** element that contains the blocking query to modify.
2. In the ePath specified for the **source** element, add an asterisk in brackets after the child object name, as shown below for the “Street” field.

```
<config>
  <block-definition number="ID1">
     <block-rule>
       <equals>
         <field>Enterprise.SystemSBR.Company.Name</field>
         <source>Company.Name</source>
       </equals>
       <range>
         <field>Enterprise.SystemSBR.CompanyAddress.Street</field>
         <source>Address[*].Street</source>
       </range>
     </block-rule>
  </block-definition>
<config>
```

3. Repeat step 2 for each child object field in the blocking query.
4. Save, validate, and close the file.

## 4.1.5 Updating the SeeBeyond Match Engine

This section describes how to update the configuration files for the standardization engine. This includes the following procedures:

- **Loading Updated Configuration Files** on page 32

- **Updating the Business Constants File** on page 33
- **Manually Updating First Name Files** on page 33

These changes are optional, but must be implemented in order to use the new features. For more information, see *Implementing the Sun SeeBeyond Match Engine with eView Studio*.

## Loading Updated Configuration Files

To use the new configuration files for the standardization engine in existing Projects, you must load them into each eView Studio Project to bring them into the Repository. You only need to perform this step if you want to use the French or Australian national domain or to update the United Kingdom or United States first name file. The files are loaded into the **Standardization Engine** node, with domain-specific files being loaded into their own subdirectories. After you upgrade eView Studio, all new files are automatically loaded into any new Projects you create.

If you update the United Kingdom or United States first name file, be aware that the differences between the 5.0.5 version and the 5.1.1 version are extensive, with many new entries, changes to existing entries, and deletions. You can use a standard diff tool to view the differences between the two versions. To do this, export the file (**personFirstNameUK.dat**) from the Project prior to loading the new file, load and then export the new file, and then compare the two files.

### To load standardization files

1. In the Project Explorer in Enterprise Designer, expand the eView Studio Project, and then expand the eView Studio application.
2. Right-click the **Standardization Engine** folder, and then select **Load Configuration Files** from the context menu.
3. In the **Open** dialog, open the folder containing the files you want to load.
4. Select the files to load.
   - To update the United Kingdom first name file, navigate to the United Kingdom subdirectory and then select **personFirstNameUK.cfg**.
   - To update the United States first name file, navigate to the United States subdirectory and then select **personFirstNameUS.cfg**.
   - To load the French domain files, select all files in the France subdirectory.
   - To load the Australian domain files, select all files in the Australia subdirectory.
5. Click **Open**.
6. On the Information dialog, click **OK**.

***Important:*** *Changes to the configuration files change how data is standardized, which can result in data that was standardized prior to using the updated files not being picked up by the blocking query after the update. In many cases the phonetic codes are still the same and would still be picked up by the query. You can write a SQL script to update any standardized or phonetically encoded data that was standardized prior to*

*updating the configuration files. Contact Sun SeeBeyond for help in writing the SQL script.*

## Updating the Business Constants File

A new parameter was added to the business constants file to define the number of tokens to allow in a business name. If no value is specified for the parameter, or it does not exist in the file, the default is the value of the **words** parameter in the **personConstants.cfg** file.

**To update the business constants file**

1. Open **bizconstants.cfg**, located in the eView Studio Project under the **Standardization Engine** node.
2. Add the following line to the parameters list in the file:

   ```
   bizMaxWords = <max_number>
   ```

   where <max_number> is the number of tokens to allow.
3. Save and close the file.

## Manually Updating First Name Files

You can manually update the standard first name files with data that is specific to your implementation. The first name files are governed by these formatting rules.

- Each entry must be in the form `original-value standardized-form gender-class`.
- Each value in the **standardized-form** column must also be listed in a different entry in the **original-value** column with a zero (0) as its standardized form. (The zero indicates the name is already standard.)

When you update the file, eView Studio internally remaps any changes that introduce inconsistencies, such as when a name is mapped to a standardized form that is mapped to a different standardized form. eView Studio maps the name to the second standardized form. For example, if a first name file contains the following entries, both “Marcus” and “Marc” are automatically mapped to “Mark” even though the standardized form for “Marcus” is specified as “Marc”.

```
Marcus   Marc     M
Marc     Mark     M
Mark     0        M
```

# 4.1.6 Updating Custom Plug-ins

This release provides improved update exception processing to prevent large stack traces from being written to the log files for custom plug-ins. In addition, log4j logging was changed to Java logging, which affects any custom plug-ins that currently call the log4j API methods. This release also provides the ability to customize the processing logic for execute match through custom plug-ins. Remember to check out the custom

plug-in files before making any changes. After you save the changes, rebuild the custom plug-ins.

### Custom Plug-in Exceptions

If a custom plug-in throws an ObjectException or SystemObjectException, multiple stack traces are logged in the server log, which can make operational management tasks more difficult. For cases where you do not want stack traces to be logged, configure the custom plug-ins to throw a UserException or one of its derived classes (DataModifiedException or ValidationException). This is useful for user errors on the Enterprise Data Manager (EDM). When one of these exceptions is thrown, no stack trace is entered in the server log but an error message still appears on the EDM.

For more information about these exception classes, see the Javadoc provided with eView Studio.

### Custom Plug-in Logging

For this release, logging for eView Studio is no longer handled using log4j, but is instead handled by Java logging. If your custom plug-ins currently call log4j API methods, replace them with the appropriate Java logging API methods. For more information, see the *Sun SeeBeyond eGate Integrator System Administration Guide*.

### Custom Plug-ins for executeMatch

This release provides the ability to customize the processing logic of the "execute match" functions in the MasterController class. In order to implement the custom logic, you must create custom plug-ins and then specify the plug-ins in the Threshold file (see **Threshold File** on page 24 for information on specifying the plug-ins). A new Java class, **ExecuteMatchLogics**, provides the methods that define the custom logic, and the execute match functions contain several different decision branches where they look for the custom plug-ins that contain the methods from the new class. For more information about customizing the match processing logic, see "Creating Custom Plug-ins" in the *Sun SeeBeyond eView Studio User's Guide*. For information about the decision branches, see Appendix A of the *Sun SeeBeyond eView Studio Reference Guide*. Additional information is also provided in **Threshold File** on page 24 and **Customizing executeMatch** on page 39 of this guide.

## 4.2 Adding an Oracle eWay to the Connectivity Map

The IS no longer supports database connectivity configuration in this release. If you are currently using the IS to connect to the database, you must reconfigure your connection through an Oracle eWay in the Connectivity Map. If you are already using the Oracle eWay to connect to the eView Studio database, you do not need to perform the following procedure, but you should verify your Oracle eWay and Oracle external system configuration properties. If you are using the Sun Java System Application Server, you can configure database connectivity through the server.

Before you begin this procedure, make sure the Oracle eWay 5.1.0 is installed.

**To add an Oracle eWay to the Connectivity Map**

1. Open the Connectivity Map in the eView Studio server Project to update.
2. On the Connectivity Map Editor toolbar, click the **External System** icon.
3. From the drop-down list, select the check box for **Oracle External Application**. An Oracle External Application icon appears on the toolbar.
4. Drag the new icon from the toolbar onto the canvas to the right of the **eView.Application-<application_name>** icon.
5. Click and drag the arrow from the right side of the **eView.Application-<application_name>** icon to the **Oracle External Application** icon.
6. Double-click the eWay icon on the joining line.
7. On the eWay Connections dialog, select **Outbound Oracle eWay** and then click **OK**.
8. On the Oracle eWay Properties window, enter the class name and database description (for more information, see the *Sun SeeBeyond eWay Adapter for Oracle User's Guide*).
9. Save and close the Connectivity Map.

## 4.3 Updating the Environment

You can either deploy the imported eView Studio Projects to an imported Environment or you can create a new Environment. For this release, several of the configuration properties that were previously set in the Enterprise Designer Environment are now set in the Enterprise Manager. For more information about these changes, see the *Sun SeeBeyond eGate Integrator User's Guide* and the *Sun SeeBeyond eGate Integrator System Administration Guide*. To create a new Environment, see the *Sun SeeBeyond eView Studio User's Guide*.

If you performed the steps under **"Adding an Oracle eWay to the Connectivity Map" on page 34**, add an Oracle external system to the Environment as described below. In addition, security for the server was previously configured in the Environment, but is now configured in the Enterprise Manager. See **Redefining Security** on page 42 for more information.

**To add an Oracle external system**

1. In Enterprise Explorer, click the **Environment Explorer** tab and then select the eView Studio Environment icon.
2. Right-click the eView Studio Environment icon to display the context menu.
3. Point to **New**, and then select **Oracle External System**.
4. In the **External System Name** field, enter the name of the new Oracle external system.
5. Click **OK**.

6 Right-click the Oracle External System icon, and then select **Properties**.

The **Properties** window appears.

7 Define the properties in the right portion of the window with information specific to the eView Studio database you created. All fields on the Properties windows are described in the *Sun SeeBeyond eWay Adapter for Oracle User's Guide*. Only define the properties for the driver type you are using.

8 When you finish defining the properties, click **OK**.

9 Map the Oracle eWay to the Oracle external system when you create the new Deployment Profile.

## 4.4 Upgrading the Database

This release includes a minor database change to improve support for multiple integration or application servers. The data type for the timestamp column of the sbyn_transaction table was changed from DATE to TIMESTAMP. Before performing this task, perform a complete backup of the eView Studio database.

***Note:*** *You must have Oracle 9.2 or later to perform this task. If you have not upgraded to Oracle 9i, do so before performing this task.*

**To update the database**

1 Connect to the eView Studio database using SQLPlus.

2 Change data type of the timestamp column in the SBYN_TRANSACTION table using the following commands.

```
ALTER TABLE SBYN_TRANSACTION MODIFY TIMESTAMP TIMESTAMP;
COMMIT;
```

## 4.5 Working With new Java API Methods

Several new Java methods are provided with this release of eView Studio to improve transaction handling and to give you more flexibility. You can add these methods to your existing Collaborations, Business Processes, or external clients. Updating your Projects with any of these methods is optional. This section provides general implementation information for the new methods. For more information about any of the new methods or classes, refer to the Javadocs provided with eView Studio.

### 4.5.1 Merging and Unmerging Records

New methods were added to the MasterController class to handle situations where one of the records involved in a merge or unmerge transaction was being updated while the merge or unmerge was being processed. The new methods are similar to the existing

merge and unmerge methods, but include the revision numbers in the parameter list to check for updates before finalizing the system object merge. The SBRs in the merge records are compared to the SBRs stored in the database and if differences are found, the merge is not allowed. (Differences indicate that either the source or destination record was modified by another user at some point during the merge process.)

A new method, **getRevisionNumber**, was added to the MasterController to retrieve the revision numbers for the new methods. The syntax for the new method is:

```
getRevisionNumber(java.lang.String euid)
```

The previous versions of the unmerge and merge methods are still available, but it is recommended that the new methods be used instead.

## New Enterprise Record Merge Methods

This release adds two new **mergeEnterpriseObject** methods to the MasterController class to better handle instances where a record is being updated at the same time the record is being merged. The syntax of the new methods is:

```
mergeEnterpriseObject(java.lang.String sourceEUID, EnterpriseObject
destinationEO, java.lang.String srcRevisionNumber, java.lang.String
destRevisionNumber, boolean calculateOnly)
```

or

```
mergeEnterpriseObject(java.lang.String fromEUID, java.lang.String
toEUID, java.lang.String srcRevisionNumber, java.lang.String
destRevisionNumber, boolean calculateOnly)
```

## New Enterprise Record Unmerge Methods

This release adds one new **unmergeEnterpriseObject** method to the MasterController class to better handle instances where a record is being updated at the same time the record is being unmerged. The syntax of the new method is:

```
unmergeEnterpriseObject(java.lang.String euid, java.lang.String
revisionNumber, boolean calculateOnly)
```

## New System Record Merge Methods

This release adds two new **mergeSystemObject** methods to the MasterController class to better handle instances where a record is being updated at the same time the system record is being merged. The syntax of the new methods is:

```
mergeSystemObject(java.lang.String systemCode, java.lang.String
sourceLID, java.lang.String destLID, ObjectNode destImage,
java.lang.String srcRevisionNumber, java.lang.String
destRevisionNumber, boolean calculateOnly)
```

or

```
mergeSystemObject(java.lang.String systemCode, java.lang.String
sourceLID, java.lang.String destLID, java.lang.String
srcRevisionNumber, java.lang.String destRevisionNumber, boolean
calculateOnly)
```

### New System Record Unmerge Methods

This release adds one new **unmergeSystemObject** method to the MasterController class to better handle instances where a record is being updated at the same time the record is being merged. The syntax of the new method is:

```
unmergeSystemObject(java.lang.String systemCode, java.lang.String
sourceLID, java.lang.String destLID, java.lang.String
srcRevisionNumber, boolean calculateOnly)
```

## 4.5.2 Deferring Potential Duplicate Calculation

This release provides new Java API methods that allow you to specify when to perform potential duplicate processing on incoming messages that result in updates. The new methods were all designed to support an override of the update mode set in the Threshold file for the eView Studio Project. If the update mode is set to pessimistic, you can use these methods to delay potential duplicate processing for updated records until a more convenient time (such as when more system resources are available). You can also use these methods to force potential duplicate processing when the update mode is set to optimistic. Below is a list of the new methods:

- MasterController.executeMatchDupRecalc
- MasterController.executeMatchUpdateDupRecalc
- MasterController.updateEnterpriseDupRecalc
- MasterController.calculatePotentialDuplicates
- MatchResult.MatchResult
- MatchResult.getMatchFieldChanged
- UpdateResult.UpdateResult
- UpdateResult.getMatchFieldChanged
- UpdateResult.setMatchFieldChanged

These methods are documented in the Javadocs provided with eView Studio. You can use these methods in the Java Collaborations, Business Processes, and any custom clients you have created. Following is an overview of how to use the methods for offline potential duplicate processing.

A call to any of the “executeMatch” functions returns a MatchResult object, which contains the Transaction number and an indicator of whether any match fields were changed during the transaction. Likewise, a call to **updateEnterpriseDupRecalc** returns an UpdateResult object, which contains a transaction results object (the EUID and Transaction number) and an indicator of whether any match fields were changed.

If any match fields were changed, store the EUID and transaction number to pass to **calculatePotentialDuplicates** when you perform potential duplicate processing. The MatchResult class provides methods to retrieve both values:

- MatchResult.getTransactionNumber - returns the transaction number
- MatchResult.getMatchFieldChanged - returns a Boolean indicator of whether a match field changed

The UpdateResult and TMResult classes provide the methods to obtain both values:

- UpdateResult.getMatchFieldChanged - returns a Boolean indicator of whether a match field changed
- UpdateResult.getTransactionResult - returns a TMResult object
- TMResult.getTMID - returns the transaction number

## 4.5.3 Customizing executeMatch

This release provides the ability to customize the processing logic of the "execute match" functions in the MasterController class. The execute match functions have been modified to contain several different decision branches where they look for logic in custom plug-ins that contain the methods from the new **ExecuteMatchLogics** class (in the package **com.stc.eindex.master**). For more information about customizing the match processing logic, see "Creating Custom Plug-ins" in the *Sun SeeBeyond eView Studio User's Guide*. For information about the decision branches, see Appendix A of the *Sun SeeBeyond eView Studio Reference Guide*. The Javadocs provided with eView Studio provide a reference for the new methods.

The following methods in **ExecuteMatchLogics** specify the custom logic.

- **bypassMatching** - indicates whether to perform the match process on incoming records or to bypass the match process
- **disallowAdd** - indicates whether an incoming message can be inserted as a new record
- **disallowUpdate** - indicates whether an incoming record can update an existing record
- **rejectAssumedMatch** - indicates whether to accept or reject an assumed match of two records
- **rejectUpdate** - indicates whether to accept or reject an update to an existing record

The custom plug-ins you create to define custom execute match logic must extend the **ExecuteProcessingLogic** class. In addition, the following classes must be imported into the custom plug-in.

- com.stc.eindex.objects.SystemObject;
- com.stc.eindex.objects.EnterpriseObject;
- com.stc.eindex.objects.exception.ObjectException;
- com.stc.eindex.master.ExecuteMatchLogics;
- com.stc.eindex.master.CustomizationException;

## 4.5.4 Using Partial Object Retrieval

This release provides a new **getEnterpriseObject** method that allows you to retrieve only the child objects you want to form the enterprise objects in a search result set, giving you control over the size of the returned records. The new method uses the ePaths of the objects you want to retrieve to form the enterprise objects. The previous

version of **getEnterpriseObject** retrieves all objects in each enterprise object. In addition, a new class, **EOGetOptions**, was created to provide the ePath parameter. The following methods are used to retrieve partial enterprise objects.

- MasterController.getEnterpriseObject
- EOGetOptions.setFieldsToRetrieve
- EOGetOptions.getFieldsToRetrieve

Below is a sample implementation of these methods. For more information, refer to the Javadocs provided with eView Studio.

```
String ePaths[] =
{
"Enterprise.SystemObject.Person.Telephone[*].*",
"Enterprise.SystemObject.Person.Address[*].*"
};
EOGetOptions options = new EOGetOptions();
options.setFieldsToRetrieve(ePaths);
EnterpriseObject eo = mc.getEnterpriseObject(euid, options);
```

## 4.5.5 Retrieving System Information

This release provides a new method, **lookupSystemDefinition**, which retrieves the attributes of an external system from the master index database given the system code. Among other uses, this allows you to substitute the system description for the system code. The syntax of the new method is:

```
SystemDefinition lookupSystemDefinition(String systemCode)
```

## 4.5.6 Updating System Objects

This release adds one new **updateSystemObject** method to the MasterController class to better handle instances where a record is being updated at the same time by two different users or processes. The syntax of the new method is:

```
updateSystemObject(SystemObject sysobj, String revisionNumber)
```

## 4.5.7 Running Assumed Match and Potential Duplicate Reports

Two new methods, **setReadForwardOnly** and **sortSummaryBy**, were added to the AssumedMatchIterator and PotentialDuplicateIterator classes. **setReadForwardOnly** specifies whether the iterator will clear the objects of a loaded page once it starts to read and load the next page. **sortSummaryBy** sorts the objects in the iterator by the specified comparator without loading the associated objects. The syntax of each new method is:

```
setReadForwardOnly(boolean forwardOnly)
```

and

```
void sortSummaryBy(String field, boolean reverse)
```

## 4.6 Completing the Project Upgrade

To ensure any changes you made take effect, perform the following tasks in the order given.

- **Regenerating the Application** on page 41
- **Updating the Collaborations** on page 41
- **Recreating the Deployment Profiles** on page 41
- **Redeploying the Projects** on page 41
- **Redefining Security** on page 42

### 4.6.1 Regenerating the Application

Before deploying the eView Studio application to the new Environment, regenerate the application to incorporate any changes you made. To regenerate, right click on the eView Studio application and then select **Generate**. Do this for each eView Studio application that was upgraded.

### 4.6.2 Updating the Collaborations

For each Collaboration client Project that references the eView Studio Java API, you must refresh the Java Collaboration to use the updated .jar files from the regenerated eView Studio server Project.

**To update the Collaborations**

1. In the Project Explorer, check out and open the Collaboration to be updated.
2. In the toolbar, click **Import JAR File**. The Add/Remove Jar Files dialog appears.
3. For each eView Studio .jar file in the list, highlight the filename and then click **Remove**.
4. For each eView Studio .jar file to re-import, do the following
   - A Click **Add**.
   - B Double-click the **eView_Sample_Collab_Client** Project name in the list that appears.
   - C Select the name of the .jar file to import.
   - D Click **Import**.

### 4.6.3 Recreating the Deployment Profiles

You must create, build, and deploy a new Deployment Profile for each upgraded Project. Deployment Profiles are described in the *Sun SeeBeyond eView Studio User's Guide*. If you added an Oracle eWay to the application, eView Studio, map the new Oracle eWay in the left pane of the Deployment Editor to the Oracle External System in the right pane.

Once all Project components are mapped to the environment, build the Projects. For information about building a Project, see the *Sun SeeBeyond eGate Integrator User's Guide* or the *Sun SeeBeyond eGate Integrator System Administration Guide*.

### 4.6.4 Redeploying the Projects

Once the eView Studio Projects are rebuilt, redeploy the Projects to the server. You can use either Enterprise Designer or Enterprise Manager to perform this task, but for the next step ("Redefining Security"), you must configure the server in Enterprise Manager. For information and instructions on redeploying Projects, see the *Sun SeeBeyond eGate Integrator User's Guide* and the *Sun SeeBeyond eGate Integrator System Administration Guide*.

### 4.6.5 Redefining Security

Any security information from version 5.0.5 must be manually defined in the Enterprise Manager. Refer to the *Sun SeeBeyond eView Studio User's Guide* for more information about the access permissions you can assign. You must connect to the server in Enterprise Manager before performing these steps.

***Note:*** *Make sure the database and domain for the eView Studio application are running before performing this procedure.*

**To create an eView Studio user account**

1. In the left frame of the Enterprise Manager, right-click the server to which you want to add a user account, and then click **Manage Integration Server Users**.

   The **Users List** window appears.

2. In the **Users List** window, click **Add New User**.

   The **Add/Edit User** window appears.

   In the **User Name** field, enter a name for the user.

3. In the **Password** field, enter a password for the user.
4. In the **Confirm Password** field, enter the password again.
5. In the **Group List** field, enter one or more eView Studio user groups, separating multiple groups with a comma. (A complete list of eView Studio user groups appears in the *Sun SeeBeyond eView Studio User's Guide*.)
6. Click **Submit**.

## 4.7 Updating Command Line Reports

If you use the command line reports, reinstall the reports to take advantage of the new configuration elements for the XML configuration files (**eIndexPersonReport.xml** and **eViewCompanyReport.xml**). These new elements are optional, but are included in the

latest version of the configuration files. They limit the number of records returned when running eView Studio reports.

- **max-result-size** - The number of records to display on a report. This element is at the same level as the **output-file** and **enable** elements for each report. If you do not enter a maximum result size or you enter "0", the size defaults to 1,000 records for all reports. To retrieve all records for a search, enter a very large value for this element.
- **status** - The status of the potential duplicate pairs to display on the Potential Duplicate report. This element is a child of the **criteria** element and is on the same level as the **dates** element. It is only supported for the Potential Duplicate report. If the status element does not exist or is left blank, potential duplicates of all statuses are returned. The value of the status element can be any of the following:
  - **U** - Unresolved
  - **A** - Auto-resolved
  - **R** - Resolved

See "Downloading eView Studio Reports" in the *Sun SeeBeyond eView Studio User's Guide* for instructions on installing the updated reports.

After you install the reports and regenerate and deploy the eView Studio server Project, export the eView Studio server Project files **<project_name>_stc_eindex_client.jar** and **<project_name>_stc_eindex_util.jar** to the **\lib** directory of the reports home directory. Rename the files to **stc_eindex_client.jar** and **stc_eindex_util.jar**.

# 4.8 Working With the EDM

For this release, the URL for accessing the Enterprise Data Manager (EDM) was changed and the file in which the timeout period is set was changed.

## Logon URL

The EDM now uses this URL for logging on: http://**<host>**:**<port>**/**<app_name>**edm, where **<host>** is the name of the server machine, **<port>** is the port number used by the EDM (by default, this is now 18001), and **<app_name>** is the name of the index application specified in the eView Studio Project. The port value to use is listed on the Domain Manager in the **HTTP** property.)

## Idle Timeout Feature

After a certain period of inactivity, the EDM automatically logs off. The default time is 30 minutes. Previously, this time period was set in the **web.xml** file located in <logicalhost_home>\stcis\catalina\conf. Now the time can be set at the server level in the **session-timeout** element of **default-web.xml** (located in <logicalhost_home>\is\domains\<domain_name>\config). It can also be set at the application level in **web.xml** in the eView Studio application .war file (located in the

deployment .ear file) or in the deployment folder itself. The application level overrides any values set at the server level.

# Index

## L



## M



## N



## O



## P



## Q



## R



## S



## T

## U



## V



## W