



Upgrading to Sun Master Index



Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 820-6014-11
October 2008

Copyright 2008 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more U.S. patents or pending patent applications in the U.S. and in other countries.

U.S. Government Rights – Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

This distribution may include materials developed by third parties.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, the Solaris logo, the Java Coffee Cup logo, docs.sun.com, Java, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. or its subsidiaries in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and SunTM Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Products covered by and information contained in this publication are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical or biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Contents

Upgrading to Sun Master Index	5
About the Upgrade Paths to Sun Master Index	6
Upgrading From eView Studio to Sun Master Index (Repository)	6
▼ To Upgrade to Sun Master Index (Repository)	6
Migrating to Service-Enabled Sun Master Index	8
Step 1: Create the Service-Enabled Master Index	8
Step 2: Migrate any Custom Plug-ins	9
Step 3: Update the Object Definition	10
Step 4: Update midm.xml	11
Step 5: Update the Master Controller	16
Step 6: Update the Matching and Standardization Configuration	17
Step 7: Update Query Definitions	17
Step 8: Update the Survivor Calculator	18
Step 9: Update Field Validations	18
Step 10: Validate the Configuration Files	19
Step 11: Update Match and Standardization Engine Files	19
Step 12: Upgrade the Database Connection and Security	21
Step 13: Finalizing the Upgrade	22
Step 14 (Optional): Update the Master Index API in Client Projects	23
Troubleshooting	23

Upgrading to Sun Master Index

The topics listed here provide procedures, conceptual information, and reference information for maintaining a Sun Master Index application.

Note that Java CAPS includes two versions of Sun Master Index. Sun Master Index (Repository) is installed in the Java CAPS repository and provides all the functionality of previous versions in the new Java CAPS environment. Sun Master Index is a service-enabled version of the master index that is installed directly into NetBeans. It includes all of the features of Sun Master Index (Repository) plus several new features, like data analysis, data cleansing, data loading, and an improved Data Manager GUI. Both products are components of the Sun Master Data Management (MDM) Suite. This document described how to upgrade to both versions.

What You Need to Know

This topic provides information you should know about upgrading to Sun Master Index.

- [“About the Upgrade Paths to Sun Master Index”](#) on page 6

What You Need to Do

These topics provide instructions on how to upgrade eView Studio and Sun Master Index (Repository).

- [“Upgrading From eView Studio to Sun Master Index \(Repository\)”](#) on page 6
- [“Migrating to Service-Enabled Sun Master Index”](#) on page 8

More Information

This topic provides additional information you should know about upgrading Java CAPS Projects.

- [Upgrading to Java CAPS 6](#)

About the Upgrade Paths to Sun Master Index

Java CAPS 6 provides two different versions of Sun Master Index — Sun Master Index (Repository) and service-enabled Sun Master Index. Sun Master Index (Repository) provides the same features and functions as eView Studio 5.1.3, with the added feature of being able to communicate with JBI and service-enabled applications. It is installed in the Java CAPS Repository. The second version, service-enabled Sun Master Index, is installed directly into NetBeans and does not use the Java CAPS Repository. It provides several new functions and tools to create a more complete data management solution.

You can upgrade an eView Studio application to a Sun Master Index (Repository) application, and you can migrate a Sun Master Index (Repository) application to a service-enabled Sun Master Index application. Upgrading from version 5.1.3 to Java CAPS 6 provides the simplest way to move your master index applications into a Java CAPS 6 environment with minimal change; however, migrating to the service-enabled version allows you to take advantage of the new features and tools.

The procedures below describe the upgrade and migration. Before you begin these steps, make sure you have installed Java CAPS 6 and have uploaded `eView.sar` to the Repository and installed it as a plug-in in NetBeans.

- [“Upgrading From eView Studio to Sun Master Index \(Repository\)”](#) on page 6
- [“Migrating to Service-Enabled Sun Master Index”](#) on page 8

Upgrading From eView Studio to Sun Master Index (Repository)

You can upgrade an eView Studio project that is at version 5.0.5 or later to a Sun Master Index (Repository) project. Before you begin this procedure, make sure you have installed Sun Master Index (Repository) as described in [“Installing Additional Non-JBI Based Java CAPS Components”](#) in *Installing Additional Components for Java CAPS 6*. Make sure to install Sun Master Index (Repository) as a NetBeans plug-in.

▼ To Upgrade to Sun Master Index (Repository)

- 1 **Export the master index client and server projects from your 5.1.3 Repository and import them into your Java CAPS 6 Repository.**

This is described in [Upgrading to Java CAPS 6](#). You can export and import the Environments along with the projects.

- 2 If you are upgrading from a version earlier than 5.1.3, update the master index project as required for the version from which you are upgrading.**

These updates are described in the *Sun SeeBeyond eView Studio Upgrade Guide, Release 5.1.3*, available at <http://docs.sun.com/app/docs/doc/820-0965>.

- 3 Regenerate the application by doing the following:**

- a. In the Projects window, expand the master index project.
- b. Right-click the master index application.
- c. On the context menu, click Generate.
- d. On the confirmation dialog box, click Yes.

- 4 Upgrade any client projects that reference the master index from Collaborations or Business Processes by following the instructions under “[Upgrading From Release 5.0.5 or Later to Release 6](#)” in *Upgrading to Java CAPS 6*.**

- 5 Update any Collaboration client projects that reference the master index server project by doing the following:**

- a. In the Projects window, check out and open the Collaboration to be updated.
- b. In the toolbar, click Import JAR File.
The Add/Remove Jar Files dialog box appears.
- c. For each Sun Master Index JAR file in the list, highlight the filename and then click Remove.
- d. For each Sun Master Index JAR file to re-import, click Add, double-click the master index service project name, select the JAR file to import, and then click import.
- e. Save the Collaboration.

- 6 If you did not import the project's Environment, recreate the Environment in Java CAPS 6.**

- 7 Create the Deployment Profile for the Master Index server project, and build and deploy the project.**

The steps for this process are described in “[Deploying the Master Index \(Repository\) Server Project](#)” in *Deploying Master Indexes (Repository)*.

- 8 Build and deploy any master index client projects.**

Migrating to Service-Enabled Sun Master Index

If you create a master index project using Sun Master Index (Repository) or if you upgrade a master index to the Repository version, you can migrate that project to the service-enabled version in order to take advantage of the new features and tools.

Follow these steps in the order given to migrate to a service-enabled master index:

- “Step 1: Create the Service-Enabled Master Index” on page 8
- “Step 2: Migrate any Custom Plug-ins” on page 9
- “Step 3: Update the Object Definition” on page 10
- “Step 4: Update midm.xml” on page 11
- “Step 5: Update the Master Controller” on page 16
- “Step 6: Update the Matching and Standardization Configuration” on page 17
- “Step 7: Update Query Definitions” on page 17
- “Step 8: Update the Survivor Calculator” on page 18
- “Step 9: Update Field Validations” on page 18
- “Step 10: Validate the Configuration Files” on page 19
- “Step 11: Update Match and Standardization Engine Files” on page 19
- “Step 12: Upgrade the Database Connection and Security” on page 21
- “Step 13: Finalizing the Upgrade” on page 22
- “Step 14 (Optional): Update the Master Index API in Client Projects” on page 23
- “Troubleshooting” on page 23

Step 1: Create the Service-Enabled Master Index

The first step to migrating a master index project is to create a service-enabled master index project. This project should be similar to the project being upgraded, but does not need to be identical because you will copy over the configuration of the project you are upgrading before you generate the new master index application.

▼ To Create the Service-Enabled Master Index

- 1 **If you are migrating from version 5.1.3 or earlier, export the master index client and server projects from your 5.1.3 Repository and import them into your Java CAPS 6 Repository.**

This is described in *Upgrading to Java CAPS 6*. You can export and import the Environments along with the projects.

- 2 **If you are upgrading from a version earlier than 5.1.3, update the master index project as required for the version from which you are upgrading.**

These updates are described in the *Sun SeeBeyond eView Studio Upgrade Guide, Release 5.1.3*, available at <http://docs.sun.com/app/docs/doc/820-0965>.

- 3 **Create a new, service-enabled master index application using the wizard, as described in “Creating a Master Index Application” in *Developing Sun Master Indexes*.**

Tip – Follow these guidelines when creating the new master index application:

- The application name and parent object name need to be the same for the new service-enabled master index as for the Repository-based master index. They also need to be the same as each other.
- You can accept the defaults for the deployment environment in the wizard since these values will be populated when you update the configuration files.
- To make it simpler, you can create the object structure using one of the predefined templates.
- For the final step of the wizard, you do not need to generate all remaining files.

- 4 **After you create the master index application, make backup copies of the configuration files. They are located in *NetBeansProjects/ProjectName/src/Configuration*.**

Step 2: Migrate any Custom Plug-ins

If any custom plug-ins were created for the project you are upgrading, you need to migrate those Java classes to the service-enabled project. You create the new custom plug-in files manually in the *ProjectName-ejb* project, and then copy the text of the existing custom plug-ins into the new files. Minor text changes are also required.

▼ To Migrate Custom Plug-ins

- 1 **For each Java package used for the custom plug-ins in the master index project you are migrating, do the following:**
 - a. **In the Projects window, expand the *ProjectName-ejb* project associated with the new service-enabled master index project.**
 - b. **In the *ProjectName-ejb* project, right-click Source Packages, point to New, and then select Java Packages.**
 - c. **On the New Java Package Wizard, enter the name of the Java package for the custom plug-in.**

Note – In Repository-based master index projects, Java package names begin with “com.stc.eindex”. In the service-enabled projects, they begin with “com.sun.mdm.index”. You should update the custom plug-in package names accordingly.

- d. Click Finish.
- 2 For each custom plug-in to be migrated, do the following:
 - a. Under Source Packages, right-click the Java package to which the custom plug-in belongs.
 - b. Point to New and then click Java Class.
 - c. On the New Java Class Wizard, enter the name of the Java class for the custom plug-in.
 - d. Click Finish.
 - e. Delete the default text in the new file.
 - f. Copy the text of the corresponding custom plug-in from the Repository-based project to the new file.
 - g. In the new file, update the package name for the new custom plug-in.
 - h. In the new file, update any package names by replacing all instances of “com.stc.eindex” with “com.sun.mdm.index”.

Note – When you first create the custom plug-ins, there will be several context errors. They should be resolved once you generate and build the master index application later in the migration process.

Step 3: Update the Object Definition

To ensure that the object definition does not change between versions, you need to copy the text of the Object Definition file in the Repository-based project to the `object.xml` file in the service-enabled project and then make certain updates to the text.

▼ To Update `object.xml`

- 1 In the service-enabled project, delete the text of `object.xml`.

- 2 **Copy and paste the text of the Object Definition file from the Repository–based master index application into `object.xml` in the service-enabled application.**
- 3 **Change the name of the `eview` element to `Configuration` (be sure to capitalize the 'C'). Change both the opening tag near the beginning of the file and the closing tag near the end of the file.**
- 4 **In the newly named `Configuration` element, change "`sbyn:SeeBeyond/eView/schema/eIndex.xsd`" to "`schema/object.xsd`".**
- 5 **Save and close the file.**

Step 4: Update `midm.xml`

The upgrade from the Enterprise Data Manager file in the Repository-based project to `midm.xml` in the service-enabled project is partially automated. A few updates to the text need to be performed manually, but the bulk of the update is done through the Configuration Editor.

▼ To Update `midm.xml`

- 1 **In the service-enabled project, delete the text of `midm.xml`.**
- 2 **Copy and paste the text of the Enterprise Data Manager file from the Repository–based master index application to `midm.xml` in the service-enabled application.**
- 3 **In the `edm` element near the top of the file, change the name of the element from "`edm`" to "`midm`", and change "`sbyn:SeeBeyond/eView/schema/EDM.xsd`" to "`schema/midm.xsd`". Be sure to change both the opening and closing `edm` tags.**

These are the lines you need to change:

```
<edm xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="sbyn:SeeBeyond/eView/schema/EDM.xsd">
...
</edm>
```

When you are done, the lines should look like this:

```
<midm xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="schema/midm.xsd">
...
</midm>
```

- 4 **Change all instances of "`com.stc.eindex`" to "`com.sun.mdm.index`".**
- 5 **Save and close the file.**

- 6 In the service-enabled project, right-click the Configuration node and then select Edit.

The Configuration Editor appears.

- 7 Enable the Save button by making a small change to the master index configuration and then changing it back.

Note – This change is just to enable the Save command so you can save `midm.xml` in the new format. For example, you can change the data type for one of the fields and then change it back to its original value.

- 8 Click Save on the Configuration Editor toolbar, and then close the Configuration Editor.

- 9 Open `midm.xml` again, and perform the following steps:

- a. Check whether any date fields have a `max-length` element with a value of `-1`. If you find fields with this value, change it to the correct value.

Tip – An easy way to check for these values is to right-click in the text of the file and then click Validate XML. If you get errors similar to The value `'-1'` of element `'max-length'` is not valid., click on the error and the cursor will move to the errant text in the file.

- b. Reformat the Reports page. Near the end of the file, delete the `reports` element and all of its contents, and then replace it with the following text.

```
<reports>
  <root-object>Person</root-object>
  <tab-name>Reports</tab-name>
  <screen-id>6</screen-id>
  <display-order>5</display-order>
  <search-pages/>
  <search-result-pages/>
  <subscreen-configurations>
    <subscreen>
      <enable>true</enable>
      <root-object>Person</root-object>
      <tab-name>Potential Duplicate Report</tab-name>
      <report-name>Potential Duplicate</report-name>
      <screen-id>0</screen-id>
      <display-order>5</display-order>
      <search-pages/>
      <search-result-pages>
        <search-result-list-page>
          <search-result-id>0</search-result-id>
          <item-per-page>10</item-per-page>
          <max-result-size>2000</max-result-size>
```

```

        <field-group>
            <description/>
            <field-ref>Person.FirstName</field-ref>
            <field-ref>Person.LastName</field-ref>
            <field-ref>Person.SSN</field-ref>
            <field-ref>Person.DOB</field-ref>
            <field-ref>Person.Gender</field-ref>
        </field-group>
    </search-result-list-page>
</search-result-pages>
</subscreen>
<subscreen>
    <enable>true</enable>
    <root-object>Person</root-object>
    <tab-name>Deactivated Record Report</tab-name>
    <report-name>Deactivated</report-name>
    <screen-id>1</screen-id>
    <display-order>1</display-order>
    <search-pages/>
    <search-result-pages>
        <search-result-list-page>
            <search-result-id>0</search-result-id>
            <item-per-page>10</item-per-page>
            <max-result-size>2000</max-result-size>
        </search-result-list-page>
    </search-result-pages>
</subscreen>
<subscreen>
    <enable>true</enable>
    <root-object>Person</root-object>
    <tab-name>Merged Transaction Report</tab-name>
    <report-name>Merged</report-name>
    <screen-id>2</screen-id>
    <display-order>0</display-order>
    <search-pages/>
    <search-result-pages>
        <search-result-list-page>
            <search-result-id>0</search-result-id>
            <item-per-page>10</item-per-page>
            <max-result-size>2000</max-result-size>
            <field-group>
                <description/>
                <field-ref>Person.FirstName</field-ref>
                <field-ref>Person.LastName</field-ref>
                <field-ref>Person.SSN</field-ref>
            </field-group>
        </search-result-list-page>
    </search-result-pages>

```

```
</subscreen>
<subscreen>
  <enable>true</enable>
  <root-object>Person</root-object>
  <tab-name>Unmerged Transaction Report</tab-name>
  <report-name>Unmerged</report-name>
  <screen-id>3</screen-id>
  <display-order>2</display-order>
  <search-pages/>
  <search-result-pages>
    <search-result-list-page>
      <search-result-id>0</search-result-id>
      <item-per-page>10</item-per-page>
      <max-result-size>2000</max-result-size>
    </search-result-list-page>
  </search-result-pages>
</subscreen>
<subscreen>
  <enable>true</enable>
  <root-object>Person</root-object>
  <tab-name>Updated Record Report</tab-name>
  <report-name>Update</report-name>
  <screen-id>4</screen-id>
  <display-order>3</display-order>
  <search-pages/>
  <search-result-pages>
    <search-result-list-page>
      <search-result-id>0</search-result-id>
      <item-per-page>10</item-per-page>
      <max-result-size>2000</max-result-size>
      <field-group>
        <description/>
        <field-ref>Person.FirstName</field-ref>
        <field-ref>Person.SSN</field-ref>
        <field-ref>Person.Address.AddressLine1</field-ref>
      </field-group>
    </search-result-list-page>
  </search-result-pages>
</subscreen>
<subscreen>
  <enable>true</enable>
  <root-object>Person</root-object>
  <tab-name>Activity Report</tab-name>
  <report-name>Transaction Summary</report-name>
  <screen-id>5</screen-id>
  <display-order>4</display-order>
  <search-pages>
    <simple-search-page>
```

```

        <screen-title>Weekly Activity</screen-title>
        <report-name>Weekly Transaction Summary Report</report-name>
        <search-result-id>0</search-result-id>
        <search-screen-order>1</search-screen-order>
        <field-group/>
    </simple-search-page>
</simple-search-page>
    <screen-title>Monthly Activity</screen-title>
    <report-name>Monthly Transaction Summary Report</report-name>
    <search-result-id>0</search-result-id>
    <search-screen-order>2</search-screen-order>
    <field-group/>
</simple-search-page>
<simple-search-page>
    <screen-title>Yearly Activity</screen-title>
    <report-name>Yearly Transaction Summary Report</report-name>
    <search-result-id>0</search-result-id>
    <search-screen-order>3</search-screen-order>
    <field-group/>
</simple-search-page>
</search-pages>
<search-result-pages>
    <search-result-list-page>
        <search-result-id>0</search-result-id>
        <item-per-page>10</item-per-page>
        <max-result-size>2000</max-result-size>
        <field-group/>
    </search-result-list-page>
</search-result-pages>
</subscreen>
<subscreen>
    <enable>>true</enable>
    <root-object>Person</root-object>
    <tab-name>Assumed Matches Report</tab-name>
    <report-name>Assumed Match</report-name>
    <screen-id>6</screen-id>
    <display-order>6</display-order>
    <search-pages/>
    <search-result-pages>
        <search-result-list-page>
            <search-result-id>0</search-result-id>
            <item-per-page>10</item-per-page>
            <max-result-size>2000</max-result-size>
            <field-group>
                <description/>
                <field-ref>Person.FirstName</field-ref>
                <field-ref>Person.LastName</field-ref>
                <field-ref>Person.SSN</field-ref>
            </field-group>
        </search-result-list-page>
    </search-result-pages>
</subscreen>

```

```
        </field-group>
    </search-result-list-page>
</search-result-pages>
</subscreen>
</subscreen-configurations>
</reports>
```

- c. **If necessary, customize the code above by replacing the values of the `field-ref` elements, replacing the value of the `root-object` elements, and configuring the items per page and maximum result size for each report.**
- d. **Verify that the screen ID and display order in the above text is not the same as the screen ID or display order for any other of the primary MIDM pages.**
- e. **Verify that there are no empty `root-object` elements. In each empty `root-object` element you find, insert the name of the parent object. For example:**

```
<root-object>Person</root-object>
```
- f. **Validate `midm.xml` by right-clicking in the text and selecting Validate XML.**

Step 5: Update the Master Controller

The Threshold file is upgraded by copying the text of the file to the service-enabled project and then updating certain portions of the file. There are only minor differences between the Threshold file in the Repository-based project and `master.xml` in the service-enabled project. In addition to updating the package name and XSD location, you can also update the file to take advantage of the new distributed transaction capabilities in Sun Master Index.

▼ To Update `master.xml`

- 1 **In the service-enabled project, delete the text of `master.xml`.**
- 2 **Copy and paste the text of the Threshold file from the Repository-based master index application to `master.xml` in the service-enabled application.**
- 3 **In the `Configuration` element near the top of the file, change "`sbyn:SeeBeyond/eView/schema/master.xsd`" to "`schema/master.xsd`".**
- 4 **Change all instances of "`com.stc.eindex`" to "`com.sun.mdm.index`".**
- 5 **After the `execute-match` element, add a new element named `transaction`. Set the value of this element to LOCAL, CONTAINER, or BEAN.**

- LOCAL means transactions are not distributed. Using this option gives you the same behavior as in previous versions.
 - CONTAINER means transaction are distributed across applications.
 - BEAN means transaction are distributed within the master index application.
- 6 Save and close the file.

Step 6: Update the Matching and Standardization Configuration

The Match Field file is updated by copying the text of the file to the service-enabled project and then updating certain portions of the file. To update the Match Field file in the Repository-based project to `mefa.xml` in the service-enabled project, you only need to change the location of the XSD file and the names of the Java packages accessed by the file.

▼ To Update `mefa.xml`

- 1 In the service-enabled project, delete the text of `mefa.xml`.
- 2 Copy and paste the text of the Match Field file from the Repository-based master index application to `mefa.xml` in the service-enabled application.
- 3 In the `Configuration` element near the top of the file, change `"sbyn:SeeBeyond/eView/schema/mefa.xsd"` to `"schema/mefa.xsd"`.
- 4 Change all instances of `"com.stc.eindex"` to `"com.sun.mdm.index"`.
- 5 Save and close the file.

Step 7: Update Query Definitions

To update the query definitions of the Repository-based project, you need to copy the text of the Candidate Select file in the Repository-based project to `query.xml` in the service-enabled project and then update certain portions of the file. For this file, you only need to change the location of the XSD file and the names of the Java packages accessed by the file.

▼ To Update `query.xml`

- 1 In the service-enabled project, delete the text of `query.xml`.

- 2 **Copy and paste the text of the Candidate Select file from the Repository–based master index application to `query.xml` in the service-enabled application.**
- 3 **In the `Configuration` element near the top of the file, change `"sbyn:SeeBeyond/eView/schema/query.xsd"` to `"schema/query.xsd"`.**
- 4 **Change all instances of `"com.stc.eindex"` to `"com.sun.mdm.index"`.**
- 5 **Save and close the file.**

Step 8: Update the Survivor Calculator

To update the Best Record file in the Repository-based project to `update.xml` in the service-enabled project, you need to copy the text of the Repository-based file to the service-enabled file and then update certain portions of the file. You only need to change the location of the XSD file and the names of the Java packages accessed by the file. If custom update policies are defined for the master index, you also need to update the Java package and class names for the policies.

▼ To Update `update.xml`

- 1 **In the service-enabled project, delete the text of `update.xml`.**
- 2 **Copy and paste the text of the Best Record file from the Repository–based master index application to `update.xml` in the service-enabled application.**
- 3 **In the `Configuration` element near the top of the file, change `"sbyn:SeeBeyond/eView/schema/update.xsd"` to `"schema/update.xsd"`.**
- 4 **If update policies are defined, scroll to the bottom of the page and update the Java package and class names of the policies to the names you gave them in [“Step 2: Migrate any Custom Plug-ins” on page 9](#).**
- 5 **Change all instances of `"com.stc.eindex"` to `"com.sun.mdm.index"`.**
- 6 **Save and close the file.**

Step 9: Update Field Validations

You only need to update the Field Validation file if you defined custom field validations in the Repository-based project. If you do need to update the file, you only need to copy the text to the service-enabled project and then change the location of the XSD file and the names of the Java packages accessed by the file.

▼ To Update validation.xml

- 1 In the service-enabled project, delete the text of `validation.xml`.
- 2 If you have defined custom field validations, copy and paste the text of the Field Validation file from the Repository-based master index application to `validation.xml` in the service-enabled application.
- 3 In the `Configuration` element near the top of the file, change `"sbyn:SeeBeyond/eView/schema/validation.xsd"` to `"schema/validation.xsd"`.
- 4 Change all instances of `"com.stc.eindex"` to `"com.sun.mdm.index"`, and update the Java package and class names for any custom plug-ins.
- 5 Save and close the file.

Step 10: Validate the Configuration Files

Once you have updated all the configuration files and used the Configuration Editor to complete the process, you should validate each file against its corresponding schema file to ensure the project can be built.

▼ To Validate the Configuration Files

- 1 Open the file in the XML editor.
- 2 Right-click in the text of the file, and select `Validate XML`.
- 3 If necessary, fix any validation errors and revalidate the file.
- 4 When the file passes validation, save and close the file.

Step 11: Update Match and Standardization Engine Files

The match engine configuration file has not changed since the previous release, and can be copied directly into the service-enabled project. Some of the standardization files were changed extensively. If you want to use the previous configuration for the person name standardization files, you will need to reformat the files.

▼ To Update Match Engine Files

- If `matchConfigFile.cfg` was customized in the Repository-based version, copy the text of the file into the service-enabled version, replacing all existing text. If the file was not customized for your implementation, you can skip this step.

▼ To Update Standardization Engine Files

- 1 If any Address or Business Name files were customized in the Repository-based project, copy the text of those files and paste it into the corresponding service-enabled files, replacing the existing text.

File names have changed as indicated in the following table. Any Address or Business Name files not listed here have the same name in both versions. Make sure you when you update the files that you update the correct country set.

Repository-Based File Name	Service-Enabled File Name
<code>addressClueAbbrev*.dat</code>	<code>clues.dat</code>
<code>addressMasterClues*.dat</code>	<code>masterClues.dat</code>
<code>addressPatterns*.dat</code>	<code>patterns.dat</code>
<code>addressConstants*.cfg</code>	Obsolete
<code>addressInternalConstants*.cfg</code>	Obsolete
<code>addressOutPatterns.dat</code>	Obsolete
<code>bizConstants.cfg</code>	Obsolete
<code>bizRemoveSpecChars.dat</code>	Obsolete

- 2 If any Person Name files were customized in the Repository-based project, do one of the following:

- **Manually make the changes to the service-enabled version.**

File names have changed for this release, as indicated in the following table. The following files are obsolete: `personNamePatt.dat`, `personConstants.cfg`, `personFirstNameDash.dat`, `personRemoveSpecChars.dat`, `personThree*.dat`, `personTwo*.dat`, `personLastNamePrefix*.dat`, and `personConjon*.dat`.

Repository-Based File Name	Service-Enabled File Name
<code>personFirstName*.dat</code>	<code>givenNameNormalization.txt</code> and <code>givenName.txt</code>

Repository-Based File Name	Service-Enabled File Name
personGenSuffix*.dat and person.OccupSuffix*.dat	generationNormalization.txt and generation.txt
personTitle.dat	titleNormalization.txt and title.txt

- **Copy the text from the Repository-based files and paste it into the corresponding service-enabled files, replacing the existing text. Make sure to copy the text into the correct country files. Then reformat the file based on the new two-column format, as shown below:**

```
STEVE|STEPHEN
STEWART|STUART
STEW|STUART
STU|STUART
SUE|SUSAN
SULLY|SULLIVAN
```

Note – For more information about the new standardization files, see “FSM–Based Person Name Configuration,” in *Understanding the Master Index Standardization Engine*.

- 3 **You can customize the following new Person Name files:** `salutationNormalization.txt`, `salutation.txt`, `surnameNormalization.txt`, and `surname.txt`.

Step 12: Upgrade the Database Connection and Security

When you migrate to service-enabled Master Index, you need to redefine the database connection pool in the application server Admin Console. You also need to set up security on the Admin Console.



Caution – As of Java CAPS Release 6 Update 1, you need to define two connection pools and two JDBC resources for Sun Master Index. For more information and instructions, see “Defining the Database Connection Pools” in *Developing Sun Master Indexes*.

▼ To Upgrade the Database Connection and Security

- 1 **Start the application server and database.**
- 2 **In the application server Admin Console, create the JDBC connection pool and data source. This procedure is described in “Defining the Database Connection Pools” in *Developing Sun Master Indexes*.**

Tip – For the connection pool, use the `javax.sql.DataSource` class for the same functionality as in previous versions. If you specified distributed transactions when you updated `master.xml`, use the `javax.sql.XADataSource` class. Once you create the Connection Pool, you can test the configuration by clicking Ping at the top of the Connection Pool page.

- 3 In the application server Admin Console, update security for existing users. Security is described in “Defining Master Index Data Manager Security” in *Maintaining Sun Master Indexes*.**
-

Note – If you created any custom user roles or groups for the Repository-based version, you can create corresponding roles in `midm-security.xml`.

Step 13: Finalizing the Upgrade

Finalizing the project includes building the project, verifying the custom plug-ins, and finally deploying the project. You can also take advantage of the new query, matching, and SBR filtering feature described in “Filtering Default Values From Master Index Processes” in *Configuring Sun Master Indexes*. Note that adding this feature might cause inconsistencies with the way records were previously matched, but can also prevent the incorrect matching of records or flagging them as potential duplicates.

▼ To Finalize the Upgrade

- 1 Right-click the Master Index project, and then click Generate Master Index Files.**
 - 2 Build the project.**

Building will compile the custom plug-ins you added previously.
 - 3 If you upgraded any custom plug-ins, go back to the custom plug-ins you created in the EJB Source Packages folder and verify that the Java code no longer has context errors.**
 - 4 If you are using an outbound Topic to broadcast master index updates, create a topic named *ApplicationNameTopic* (where *ApplicationName* is the name of the master index application) and set up the external systems to subscribe to this topic.**
-

Note – For more information about this process, see [Broadcasting Master Index Updates to External Systems](#).

- 5 If you created any new user roles in `midm-security.xml` or `security.xml`, add those roles to `web.xml` and `sun-web.xml`. These files are located in the *ProjectName-war* project under Web Pages > WEB-INF.**

- 6 Deploy the master index project.

Step 14 (Optional): Update the Master Index API in Client Projects

Once you build the master index project, a set of dynamic API methods are generated for use in web services, Collaborations, and Business Processes. In order to make the API available to Repository-based client projects, you need to generate a WSDL file and then import the generated file into the client project that will access the new service-enabled master index. Alternatively, you can upgrade the client projects using JBI components and call master index services from BPEL processes, web services, and Java clients.

▼ To Update the Master Index API in Client Projects

- 1 In the NetBeans Projects window, expand the *ProjectName-ejb* project and then expand Web Services.
- 2 Right-click *ApplicationNameEJBService* and then select Generate and Copy WSDL.
- 3 Specify a location to store the WSDL file and click OK.
- 4 Right-click the Repository-based client project, point to Import, and then select Web Service Definition.
- 5 Follow the steps in the Import WSDL(s) Wizard.

A web service OTD for the *ProjectNameEJBService* is created. You can use this OTD to access the operations of the migrated master index application.

Troubleshooting

If you use the Configuration Editor to modify the master index application once all of the Repository-based information has been copied over, there might be invalid XML code in `update.xml` and `meta.xml`.

Standardization and Normalization

Standardization and normalization structures might get removed from `meta.xml`. If you use the Configuration Editor, be sure to check `meta.xml` for any discrepancies.

Survivor Strategy

The Configuration Editor might make changes to the survivor strategy in `update.xml` that prevent the project from being built. To fix the file, do the following:

1. Open `update.xml`.
2. If any candidate fields have a specific strategy defined, reformat the XML structure for that candidate field by removing the forward slash at the end of the `candidate-field` element and adding a closing `candidate-field` element after the `survivor-strategy` element.

For example, the following excerpt:

```
<candidate-field name="Person.Phone[*].*" />
  <survivor-strategy>
    <strategy-class>com.sun.mdm.index.survivor.impl.WeightedSurvivorStrategy
    </strategy-class>
    <parameters>
      <parameter>
        <parameter-name>ConfigurationModuleName</parameter-name>
        <parameter-type>java.lang.String</parameter-type>
        <parameter-value>WeightedSurvivorCalculator</parameter-value>
      </parameter>
    </parameters>
  </survivor-strategy>
```

Would change to this:

```
<candidate-field name="Person.Phone[*].*">
  <survivor-strategy>
    <strategy-class>com.sun.mdm.index.survivor.impl.WeightedSurvivorStrategy
    </strategy-class>
    <parameters>
      <parameter>
        <parameter-name>ConfigurationModuleName</parameter-name>
        <parameter-type>java.lang.String</parameter-type>
        <parameter-value>WeightedSurvivorCalculator</parameter-value>
      </parameter>
    </parameters>
  </survivor-strategy>
</candidate-field>
```