



# Broadcasting Master Index Updates to External Systems



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

Part No: 820-5701-13  
September 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 Sun<sup>TM</sup> 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

---

<b>Broadcasting Master Index Updates to External Systems</b> .....	5
Related Topics .....	6
About Sun Master Index Broadcasts .....	6
Creating a JMS Connection Factory .....	7
▼ To Create a JMS Connection Factory .....	7
Defining the Client Connection to the JMS Topic Using BPEL .....	7
Creating the BPEL Project for the Master Index Topic .....	8
Creating a Web Service Definition for the JMS Binding Component .....	8
Creating a Web Service Definition for the File Binding Component .....	10
Creating a BPEL Process to Access the Master Index Topic .....	11
Creating a Composite Application for Master Index Broadcasts .....	14
▼ To Create the Composite Application .....	14
Testing the Master Index Broadcast Configuration .....	14
▼ To Test the Master Index Broadcast Configuration .....	15
Sample Master Index Broadcast Message .....	15



# Broadcasting Master Index Updates to External Systems

---

The topics listed here provide procedures, conceptual information, and reference information for broadcasting Sun Master Index updates to the external systems that share data with the 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 relates to Sun Master Index only.

## **What You Need to Know**

These topics provide information you should know about broadcasting master index application update messages.

- [“About Sun Master Index Broadcasts” on page 6](#)
- [“Sample Master Index Broadcast Message” on page 15](#)

## **What You Need to Do**

These topics provide instructions on how to configure a Topic to broadcast master index application updates, and how to create client projects that subscribe to the Topic.

- [“Creating a JMS Connection Factory” on page 7](#)
- [“Defining the Client Connection to the JMS Topic Using BPEL” on page 7](#)
- [“Creating a Composite Application for Master Index Broadcasts” on page 14](#)
- [“Testing the Master Index Broadcast Configuration” on page 14](#)

## Related Topics

Several topics provide information and instructions for implementing and using a master index application. For a complete list of topics related to working with the service-enabled version of Sun Master Index, see [“Related Topics” in \*Developing Sun Master Indexes\*](#).

## About Sun Master Index Broadcasts

A master index application can be configured to broadcast any updates made to the database to the external systems that share data with the master index application. These updates can be a result of messages sent in from external systems or of changes made directly through the Master Index Data Manager (MIDM). Configuring the master index application to broadcast updates provides external systems with access to the most current and accurate information about the entities being indexed. Note that the external systems must be capable of accepting incoming messages in order to receive the broadcasts.

Sun Master Index uses a JMS Topic to broadcast updates. A JMS Topic is a JMS provider that uses publish and subscribe messaging. This means that a single publisher broadcasts messages to multiple subscribers, ensuring that all subscribers receive the messages. The update message are created in XML format; the schema file that defines the structure of the message is generated when you generate the master index application files. It is located at `NetBeansProjects/ProjectName/files-generated/outbound.xsd`.

To set up a master index application to broadcast updates to external systems, you need to create a connection factory, which is used to connect to the Topic. Once the factory is created, you can define a BPEL process or other JMS client to subscribe to the Topic for each external system. This document provides instructions for using a BPEL process.

Follow these steps in the order given to configure a master index application to broadcast update messages:

- [“Creating a JMS Connection Factory” on page 7](#)
- [“Defining the Client Connection to the JMS Topic Using BPEL” on page 7](#)
- [“Creating a Composite Application for Master Index Broadcasts” on page 14](#)
- [“Testing the Master Index Broadcast Configuration” on page 14](#)

---

**Tip** – Before you begin these steps make sure the master index application has been generated and deployed, and make sure the application server and database are running.

---

## Creating a JMS Connection Factory

In order to use a Topic with a master index application, you need to create a JMS connection factory using the application server Admin Console. Creating the connection factory automatically creates a connector connection pool and resource for the factory. The connection factory must have the name `jms/ApplicationNameOutBoundSender`.

### ▼ To Create a JMS Connection Factory

**Before You Begin** Make sure the application server is running.

- 1 **Launch the application server Admin Console.**
- 2 **In the left panel of the Admin Console, expand Resources, expand JMS Resources, and then expand Connection Factories.**

The JMS Connection Factories page appears.

- 3 **In the Connection Factories table, click New.**

The New JMS Connection Factory page appears.

- 4 **Enter the JNDI name for the connection factory in the following format:**

`jms/ApplicationNameOutBoundSender`

Where *ApplicationName* is the name of the master index application whose updates are being broadcast. Note that this name is case sensitive. The application name must match the case of the actual master index application, and the suffix “OutBoundSender” must be capitalized as shown.

- 5 **In the Resource Type field, select `javax.jms.TopicConnectionFactory`.**
- 6 **Click OK.**
- 7 **Continue to [“Defining the Client Connection to the JMS Topic Using BPEL”](#) on page 7.**

## Defining the Client Connection to the JMS Topic Using BPEL

You can use a BPEL process or other JMS client (such as a Java client, JMS Binding Component, or message driven bean) to subscribe to the messages that the master index application sends to the JMS Topic. The BPEL process or JMS client can then forward the messages to the appropriate external systems. The instructions provided here describe how to send the messages to a flat file in XML format using a BPEL process. In a production environment, the File binding type would likely be replaced by HTTP or SOAP.

- “Creating the BPEL Project for the Master Index Topic” on page 8
- “Creating a Web Service Definition for the JMS Binding Component” on page 8
- “Creating a Web Service Definition for the File Binding Component” on page 10
- “Creating a BPEL Process to Access the Master Index Topic” on page 11

## Creating the BPEL Project for the Master Index Topic

The BPEL project contains the BPEL process and web service definitions that define how to process the messages published by the master index Topic.

### ▼ To Create the BPEL Project

- 1 Complete the procedure described in [“Creating a JMS Connection Factory” on page 7](#).
- 2 In the NetBeans toolbar, click **New Project**.  
The New Project Wizard appears.
- 3 In the **Categories** box, select **CAPS > ESB**.
- 4 In the **Projects** box, select **BPEL Module**, and then click **Next**.
- 5 Enter the name and location of the new project, and then click **Finish**.
- 6 Continue to [“Creating a Web Service Definition for the JMS Binding Component” on page 8](#).

## Creating a Web Service Definition for the JMS Binding Component

A JMS BC provides the connection between the JMS Topic and the BPEL process. The web service definition configures this connection.

### ▼ To Create the JMS Web Service Definition

- 1 Complete the procedure described in [“Creating the BPEL Project for the Master Index Topic” on page 8](#).
- 2 Right-click the new BPEL project, point to **New**, and then select **WSDL Document**.  
The New WSDL Document Wizard appears.
- 3 In the upper portion of the window, enter a file name and folder name for the WSDL file.

- 4 **In the lower portion of the window, do the following:**
  - a. **Select Concrete WSDL Document.**
  - b. **In the Binding Field, select JMS.**
  - c. **In the Type field, select Receive.**
- 5 **Click Next.**

The Request Connection Configuration window appears.
- 6 **In the JMS Connection section, enter the following information:**
  - **Connection URL** – The URL to connect to the message server. The format of the URL is `mq://host_name:port`. The default port is 7676.
  - **User Name** – The user name to log on to the message server. The default user name is admin.
  - **Password** – The password to log on to the message server. The default password is admin.
- 7 **(Optional) Configure the Payload Processing section according to your requirements.**
- 8 **Click Next.**

The Request Consumer Configuration window appears.
- 9 **Configure the Destination Properties as follows:**
  - a. **For the Destination property, enter *Application\_Name*Topic, where *Application\_Name* is the name of the master index application.**
  - b. **For the Destination Type, select Topic.**
  - c. **For Subscription Durability, select Non-durable.**
  - d. **Select XA Transaction if you are using XA transactions.**
- 10 **In the JMS Consumer Properties, you can leave the default or empty values.**
- 11 **Click Next.**

The Advanced Configuration window appears.
- 12 **You can enter any of the advanced options listed at the bottom of this window, or click Finish if you do not need advanced options.**

The contents of the WSDL file appears in the editor after you click Finish.

## Creating a Web Service Definition for the File Binding Component

For testing purposes, you can publish the master index broadcasts to a file using the File BC as described below. In a production environment, it is more likely that a SOAP or HTTP binding type would be used.

### ▼ To Create the File Web Service Definition

- 1 Complete the procedure described in [“Creating a Web Service Definition for the JMS Binding Component” on page 8](#).
- 2 **Right-click the BPEL project, point to New, and then select WSDL Document.**  
The New WSDL Document Wizard appears.
- 3 **In the upper portion of the window, enter a file name and folder for the WSDL file.**
- 4 **In the lower portion of the window, do the following:**
  - a. **Select Concrete WSDL Document.**
  - b. **In the Binding Field, select File.**
  - c. **In the Type field, select Write.**
- 5 **Click Next.**  
The Request Configuration window appears.
- 6 **In the File Write section, enter the following information:**
  - **File Name** – The name of the file that output data will be written to.
  - **Path** – The absolute path to the directory where you want to store the output file.
- 7 **(Optional) Configure the Payload Processing section according to your requirements.**
- 8 **Click Finish.**  
The contents of the WSDL file appears in the editor.
- 9 **Continue to [“Creating a BPEL Process to Access the Master Index Topic” on page 11](#).**

## Creating a BPEL Process to Access the Master Index Topic

The BPEL process defines the flow of data from the JMS Topic to the external systems that share data with the master index application using the web service definitions you created earlier. The BPEL process created below uses a File binding type. Your process might differ from this based on the binding type you use.

Perform the following steps to create and configure the BPEL process:

- “To Create the BPEL Process” on page 11
- “To Create Partner Links” on page 11
- “To Create a Receive Activity” on page 12
- “To Create an Invoke Activity” on page 13
- “To Create an Assign Activity” on page 13

### ▼ To Create the BPEL Process

- 1 Complete the procedure described in [“Creating a Web Service Definition for the File Binding Component” on page 10](#).
- 2 In the Projects window, right-click the BPEL project, point to New, and then select BPEL Process. The New BPEL Process Wizard appears.
- 3 Enter a file name and location for the BPEL Process, and then click Finish. The business process appears in the Business Process Editor.
- 4 Continue to [“To Create Partner Links” on page 11](#).

### ▼ To Create Partner Links

- 1 Create a new Partner Link for the JMS web service definition by doing the following:
  - a. In the BPEL Designer Palette, click Partner Link and drag it to the drop point on the left. The Create New Partner Link window appears.
  - b. Fill in the information for the Partner Link.
    - Enter a unique name for the Partner Link.
    - For the WSDL file, select the JMS Binding Component WSDL file you created in [“Creating a Web Service Definition for the JMS Binding Component” on page 8](#).

- Select the option button next to Use Existing Partner Link Type.
  - Accept the default values for the Partner Link Type, My Role, and Partner Role fields.
- c. Click OK.
- 2 Create a new Partner Link for the File web service definition by doing the following:
- a. In the BPEL Designer Palette, click Partner Link and drag it to the drop point on the right. The Create New Partner Link window appears.
  - b. Fill in the information for the Partner Link.
    - Enter a unique name for the Partner Link.
    - For the WSDL file, select the File Binding Component WSDL file you created in [“Creating a Web Service Definition for the File Binding Component”](#) on page 10.
    - Select the option button next to Use Existing Partner Link Type.
    - Accept the default value for the Partner Link Type, My Roles, and Partner Role fields.
  - c. Click OK.
- 3 Continue to [“To Create a Receive Activity”](#) on page 12.

## ▼ To Create a Receive Activity

- 1 From the BPEL Designer Palette, drag a Receive activity to the drop point.
- 2 Click the Edit icon above the new Receive activity.
- 3 Enter a name for the activity.
- 4 In the Partner Link field, select the JMS Partner Link you created earlier.
- 5 In the Operation field, select the JMS subscriber operation.
- 6 Next to the Input Variable field, select Create. The New Input Variable dialog box appears.
- 7 Enter a name for the variable and select the current BPEL process as the Scope.

- 8 Click OK on both dialog boxes.
- 9 Continue to [“To Create an Invoke Activity” on page 13.](#)

### ▼ **To Create an Invoke Activity**

- 1 From the BPEL Designer Palette, drag an Invoke activity to the drop point beneath the Receive activity just created.
- 2 Click the Edit icon above the new Invoke activity.
- 3 Enter a name for the activity.
- 4 In the Partner Link field, select the File Partner Link you created earlier.
- 5 In the Operation field, select the File operation.
- 6 Next to the Input Variable field, select Create.  
The New Input Variable dialog box appears.
- 7 Enter a name for the variable and select the current BPEL process as the Scope.
- 8 Click OK on both dialog boxes.
- 9 Continue to [“To Create an Assign Activity” on page 13.](#)

### ▼ **To Create an Assign Activity**

- 1 From the BPEL Designer Palette, drag an Assign activity to the drop point between the Receive and Invoke activities.

---

Tip – The Assign activity is listed under Basic Activities in the Palette.

---

- 2 **Double-click the Assign activity.**  
The BPEL Mapper appears.
- 3 **Map the input value from the Receive activity's variable to the Invoke activity's variable.**
- 4 **Save and close the BPEL Process.**
- 5 **Continue to [“Creating a Composite Application for Master Index Broadcasts” on page 14.](#)**

# Creating a Composite Application for Master Index Broadcasts

The Composite Application project creates a service assembly, which can then be deployed to the application server. This project assembles the BPEL process and web service definitions you created earlier into an application.

## ▼ To Create the Composite Application

- 1 Complete the procedures described in [“Creating a BPEL Process to Access the Master Index Topic” on page 11](#)
- 2 Right-click in the NetBeans Projects window, point to New, and then select Project.  
The New Project Wizard appears.
- 3 Under Categories, select CAPS > ESB.
- 4 Under Project, select Composite Application.
- 5 Enter a name and location for the project.
- 6 Click Finish.
- 7 Drag the BPEL project from the Projects window and drop it in the JBI Modules section of the CASA Editor canvas.
- 8 In the CASA Editor toolbar, click Build Project.
- 9 After the project builds successfully, click Deploy in the CASA Editor toolbar.
- 10 Continue to [“Testing the Master Index Broadcast Configuration” on page 14](#).

## Testing the Master Index Broadcast Configuration

Once you have completed all the steps for configuring the master index application to broadcast updates, you can test the system to be sure it is working. You can view a sample broadcast message in [“Sample Master Index Broadcast Message” on page 15](#)

## ▼ To Test the Master Index Broadcast Configuration

**Before You Begin** Make sure the master index project is deployed and the database is running.

- 1 Complete the procedure described in [“Creating a Composite Application for Master Index Broadcasts” on page 14.](#)
- 2 Launch the MIDM.

---

**Tip** – The URL for the MIDM is `http://localhost:port/ApplicationNameMIDM`, where *port* is the HTTP port number of the application server (8080 by default) and *ApplicationName* is the name of the master index application. The application name and “MIDM” are case sensitive.

---

- 3 From the **Source Records Add** page, create a new record in the master index database.
- 4 On your computer, navigate to the location you specified for the **File Binding Component** web service definition in [“Creating a Web Service Definition for the File Binding Component” on page 10.](#)
- 5 Open the XML file and view the contents.

## Sample Master Index Broadcast Message

The outbound messages generated by updates to the master index application include several transactional fields, system and local ID information, and the record’s SBR. The SBR provides external systems with the single best data for a particular entity along with the EUID.

Transactional fields include information such as the update function, system, user, and date; create function and user; and status. The local ID and system information provides a list of system and local ID pairs that are associated with the record, allowing you to easily identify which records in the external systems are affected by the update.

A unique schema definition file (XSD file) is generated for each master index application based on the object structure defined for that application. You can view the XSD file at `NetBeansProjects/ProjectName/files-generated/outbound.xsd`.

Below is a sample broadcast message for EUID 0000006501, which is associated with three different system records. The master index application is based on the Person template.

```
<?xml version="1.0" encoding="UTF-8"?>
<OutMsg Event="UPD" ID="00000000000000004010">
  <SBR EUID="0000006501" Status="active" CreateFunction="Add" ChildType="Customer"
    CreateSystem="System" UpdateFunction="Update" RevisionNumber="3" CreateUser="mdm"
    UpdateSystem="System" UpdateDate="" CreateDate="" UpdateUser="mdm">
    <SystemObject SystemCode="ORACLE" LID="2372385720" Status="active"></SystemObject>
```

```

<SystemObject SystemCode="ORACLE" LID="2837482562" Status="active"></SystemObject>
<SystemObject SystemCode="SAP" LID="8327423434" Status="active"></SystemObject>
<Customer PersonCatCode="MEMBER" FirstName="ELIZABETH" FirstName_Std="ELIZABETH"
  FirstName_Phon="E421" MiddleName="ANN" LastName="WARNER" LastName_Std="WARNER"
  LastName_Phon="WARNAR" Suffix="" Title="DR" SSN="555999222"
  DOB="04/14/1964 00:00:00" Death="" Gender="F" MStatus="M" Race="C" Ethnic="31"
  Religion="AG" Language="EN" SpouseName="CRAIG" MotherName="JEN" MotherMN="SMITH"
  FatherName="MARK" Maiden="MILLER" PobCity="MILFORD" PobState="ONTARIO"
  PobCountry="CAN" VIPFlag="NONE" VetStatus="NONE" Status=""
  DriverLicense="CT12941284" DriverLicenseSt="CT" Dod="" DeathCertificate=""
  Nationality="CAN" Citizenship="CAN">
  <Alias FirstName="LIZ" MiddleName="" LastName="MILLER"></Alias>
  <Address AddressType="H" AddressLine1="1330 BLOSSOM ST."
    AddressLine1_HouseNo="1330"AddressLine1_StDir="" AddressLine1_StName="BLOSSOM"
    AddressLine1_StPhon="BLASAN" AddressLine1_StType="St" AddressLine2=""
    City="SHEFFIELD" StateCode="CT" PostalCode="09876" PostalCodeExt=""
    County="CAPE BURR" CountryCode="USA"></Address>
  <Phone PhoneType="CC" Phone="9894774477" PhoneExt=""></Phone>
</Customer>
</SBR>
</OutMsg>

```