Using the Sun Data Mashup Engine

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

Part No: 820–6013
December 2008

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.

081223@21808

Contents

Using the Sun Data Mashup Engine

The topics listed here provide instructions on how to use the Sun™ Data Mashup Engine, a JBI-compliant service engine that provides Enterprise Data Federation or mashup services. It provides data services to the Sun Java™ Composite Application Platform Suite (Java CAPS) components, included with Java CAPS 6 Update 1, Sun GlassFish Enterprise Service Bus (GlassFish ESB), and the Master Data Management (MDM) Suite.

Data Mashup is automatically installed with a complete Java CAPS 6 Update 1 or Sun GlassFish ESB installation. See "Data Mashup Prerequisites" on page 7 for details on what must be installed before you can use Data Mashup.

If you have any questions or problems and have purchased a support contract, see the Java CAPS support web site at `http://goldstar.stc.com/support`.

Note – Requires Java CAPS 6 Update 1 or Sun GlassFish ESB.

What You Need to Know

These topics provide conceptual information.

- "Sun Data Mashup Engine Overview" on page 6
- "Sun Data Mashup Engine Application Configuration" on page 26
- "Sun Data Mashup Engine REST Support" on page 27
- "Known Data Mashup Issues" on page 29

To access all the Java CAPS documentation in HTML format on the Sun Developer Network (SDN) web site, go to `http://developers.sun.com/docs/javacaps/`.

What You Need to Do

These topics provide instructions on how to use the Data Mashup Engine to create, configure, and deploy Data Mashup projects.

- "Configuring and Starting the Sun GlassFish Enterprise Server" on page 7

- “Designing Data Mashup Projects” on page 9
- “Configuring Data Mashup Projects Using Joins” on page 16
- “Creating and Deploying a Composite Application to Complete a Data Mashup Project” on page 18
- “Testing the Deployment of a JBI Composite Application” on page 21
- “Viewing the Data” on page 23

Sun Data Mashup Engine Overview

Use the Sun Data Mashup Engine to build a project with unified views of your data from different sources, configure the project, create and deploy a composite application, and enable the generated report to function as a virtual database. Data Mashup is a standalone product. It can, however, expose certain JBI-based MDM Suite data sources as services.

Note – Requires Java CAPS 6 Update 1 or Sun GlassFish ESB.

Data Mashup is an engine that provides server side mashup for dispersed data. More specifically, Data Mashup provides a single relational view of data that have different origins but are located within the same enterprise. You use the service engine to integrate information from delimited flat files, fixed width flat files, relational databases, RSS Feeds, web (HTML), XML, WebRowSet, and MS Excel spreadsheets, and then join the various data sources, cleanse the data, and generate a report. After a successful deployment of the composite application, which contains the Data Mashup project, you can open the generated report in a browser, where it functions as a web service and can consume other applications. The steps to complete this task include:

- “Designing Data Mashup Projects” on page 9
- “Configuring Data Mashup Projects Using Joins” on page 16
- “Creating and Deploying a Composite Application to Complete a Data Mashup Project” on page 18
- “Testing the Deployment of a JBI Composite Application” on page 21
- “Viewing the Data” on page 23

Note – Java CAPS projects that have non-JBI components and a WSDL can use the JBI Bridge to use Data Mashup.

Data Mashup is automatically installed with a complete Java CAPS R6 U1 or Sun GlassFish ESB installation. See “Data Mashup Prerequisites” on page 7 for details on what must be installed before you can use Data Mashup.

Data Mashup Prerequisites

Before you can use the Data Mashup Engine, you must install:

- NetBeans™ IDE 6.1
- Sun GlassFish Enterprise Server 9.1 U2
- Sun Data Mashup Engine (JBI Runtime)
- Sun Data Mashup Engine Design-time environment NetBeans IDE modules

All four of these are installed by Java CAPS R6 U1 or Sun GlassFish ESB.

See Also

- Using the Sun Data Mashup Engine
- "Sun Data Mashup Engine Application Configuration" on page 26
- "Sun Data Mashup Engine REST Support" on page 27
- "Known Data Mashup Issues" on page 29

Configuring and Starting the Sun GlassFish Enterprise Server

This topic provides information for configuring and then starting the Sun GlassFish Enterprise Server.

Note – Requires Java CAPS 6 Update 1 or Sun GlassFish ESB.

While working on creating a Data Mashup project in the NetBeans IDE you do not need to have the Sun GlassFish Enterprise Server running.

▼ To Start the Sun GlassFish Enterprise Server

This topic leads you through the steps to start the Sun GlassFish Enterprise Server.

Note – You do not need the application server running all the time. When Data Mashup requires the application server, it automatically starts it.

1 **To start the NetBeans IDE, navigate to the appropriate directory and start the executable file.**

Platform	Directory Path	Executable
Solaris, AIX, and Linux	`GlassFish ESB-install-directory/netbeans/bin`	`./netbeans.sh`
Windows	`GlassFish ESB-install-directory\netbeans\bin`	`netbeans.exe`

2 **Start the Sun GlassFish Enterprise Server in the NetBeans IDE.**

a. **Select the Services tab.**

b. **Expand the Servers node.**

c. **Right-click GlassFish V2 and select Start.**

Wait until the message, "Application server startup complete," appears in the Output window and a green arrow points at GlassFish V2 in the runtime window.

3 **Start the EDM Engine.**

a. **Expand the GlassFish V2 node in the runtime window.**

b. **Expand the JBI node.**

c. **Expand Service Engines.**

d. **Right–click** `sun-edm-engine` **and select Start.**

Next Steps You are now ready to begin designing a Data Mashup project. For specific instructions on how to do this see "Designing Data Mashup Projects" on page 9.

See Also

- "Configuring Data Mashup Projects Using Joins" on page 16
- "Creating and Deploying a Composite Application to Complete a Data Mashup Project" on page 18
- "Testing the Deployment of a JBI Composite Application" on page 21
- "Viewing the Data" on page 23

For additional information, see:

- Using the Sun Data Mashup Engine

Designing Data Mashup Projects

The following procedure leads you through the necessary steps to design a Data Mashup project, and includes:

- Creating a virtual database
- Adding a minimum of two types of external data, for example a spreadsheet and a text file
- Creating a Data Mashup project

Note – Requires Java CAPS 6 Update 1 or Sun GlassFish ESB.

For this procedure you are going to create a federated join using two data files, a MS Excel spreadsheet (`company-data.xls`) and a delimited text file (`supplier-address.txt`). Ensure that you have your data files accurately set up before beginning the procedure. The files used in the procedure include the following information.

TABLE 1 Contents of `company-data.xls` Spreadsheet

EQ162	Funnel	Shelly glass and labs	LC05647
EQ763	Spirit	Torry Harris Equipment	LC23456
EQ785	Beaker	Torry Harris Equipment	LC23456
EQ232	Wire mesh	Angels Enterprises	LC34267
EQ892	Stirrer	Stirrer Angels Enterprises	LC34267
EQ232	Wire mesh	Kittens Inc.	LC45634
EQ763	Asbestos sheets	Labkron Instruments	LC56473
EQ785	Beaker	Labkron Instruments	LC56473
EQ162	Funnel	Labtech Inc.	LC56743
EQ763	Asbestos sheets	Labtech Inc.	LC56743
EQ763	Spirit	Peter Labs and Instruments	LC63234
EQ563	Test tube	Peter Labs and Instruments	LC63234
EQ092	Pipette	Patel Lab Exports	LC67384
EQ243	Bunsen burner	Alaska Lab Equipment	LC76483
EQ892	Stirrer	Alaska Lab Equipment	LC76483

TABLE 1 Contents of `company-data.xls` Spreadsheet *(Continued)*

EQ162	Funnel	Safe Labs Inc.	LC87635
EQ563	Test tube	Davidson Equipment Exports	LC87645
EQ092	Pipette	Andrew Instruments	LC97627
EQ243	Bunsen burner	Andrew Instruments	LC97627
EQ162	Funnel	Andrew Instruments	LC97627

TABLE 2 Contents of `supplier-address.txt` Delimited Text File

LC76434|Yahoo Labs and Instruments|1285, 1st Avenue 69th and 70th, New York, NY 10021

LC76323|Isabel Instruments|330 SUNRISE HIGHWAY, ROCKVILLE CTR, NY 11570

LC76483|Alaska Lab Equipment|275 SOUTH STREET, OYSTER BAY, NY 11771

LC63234|Peter Labs and Instruments|430 E MAIN ST, BAY SHORE, NY 11706

LC34267|Angels Enterprises|2950 AVENTURA BLVD, AVENTURA, FL 33180

LC45634|Kittens Inc.|South Town Center, 10450 S State St, Sandy, UT 84070

LC87635|Safe Labs Inc.|Guadalupe Center, 333 Montezuma Ave, Santa Fe, NM 87501

LC56473|Labkron Instruments|Casis Village, 2700 Exposition, Austin, TX 78703

LC56743|Labtech Inc.|Midtown, 911 W 38th St, Austin, TX 78705

LC67384|Patel Lab Exports|3 University Dr, Augusta, ME 04330

LC05647|Shelly glass and labs|633 W 5th St Los Angeles, CA 90071

LC87645|Davidson Equipment Exports|South Town Center, 10450 S State St, Sandy, UT 84070

LC23456|Torry Harris Equipment|Fairlawn Drive Up, 5325 Sw 21st St, Topeka, KS 66604

LC97627|Andrew Instruments|Valero 4550, 115 N Greeley Hwy, Cheyenne, WY 82001

The spreadsheet and delimited text file can be downloaded from `http://www.esnips.com/web/DataMashupSandraUseCaseFiles`. Note that the name of the `*.xls` file that is available to download has been changed from `Sandra-data.xls` to `company-data.xls` in the procedure that follows.

▼ To Design a Data Mashup Project

Before You Begin Before beginning this procedure, you must have installed the components listed in "Sun Data Mashup Engine Overview" on page 6, have the above-mentioned files ready and stored on your computer, and have started the NetBeans IDE.

Note – To move forward or backward in the procedure, click Next or Back.

1 **Create the Data Mashup Database.**

a. **From the NetBeans IDE tool bar select Tools→Virtual Database→Create Virtual Database.**

b. **In the Create Virtual Database window, type a Database Name that represents the project you are creating.**

In this procedure the database is called `VirtualMashupDB`.

c. **After naming the database, click Finish.**

A message confirms that you created the database successfully.

2 **Add external data to the Virtual Database.**

Note – As stated in Before You Begin, two data sources are used, a spreadsheet and a delimited text file. You are going to merge them into one table. For this step add the spreadsheet to the table, and then fine-tune some of its fields.

There are seven substeps that must be completed to add an external table.

a. **From the NetBeans IDE tool bar, select Tools→Virtual Database→Add External Tables.**

b. **In the Add External Tables/Choose Virtual Database window, ensure that the appropriate database from the list of available databases is highlighted before continuing.**

In the example, `VirtualMashupDB_1222123456789`, which is the database you just created. The underscore–numerical extension has been automatically added to the database name. The extension guarantees that you cannot create two databases with the same name, as no two extensions are the same.

c. **In the Add External Tables/Choose Data source window, navigate to the directory that contains the spreadsheet, select it, and click Add.**

In this procedure use `company-data.xls`. The file now appears in two locations under the Choose Data source pane: File name and Selected Table Source. You can remove it by highlighting the file name in the Selected Table Source and clicking Remove. When you are satisfied that your selection is correct, continue with the procedure.

Note – Although you are selecting a file that is located in a local directory, you can also select data located on the web by entering the URL.

d. **In the Add External Tables/Enter Table Details window, confirm that the table name, encoding, table type, and resource URL are correct.**

Note – The resource URL is the directory path and file name of the spreadsheet you selected.

e. **In the Add External Tables/Choose a Sheet window, click Preview to view the spreadsheet you added to the database.**

In the example, you used `company-data.xls`, which is Sheet1. If this sheet is not the file you want to work with, select another sheet.

Note – After building the table you can change the URL if desired.

For this creation of a Data Mashup project, skip step 5, Choose a (HTML) Table.

f. **In the Add External Tables/Import Table MetaData window, set how you want to parse the file.**

If the data in the preview does not contain column names, and in the example it does not, deselect the First line contains field names? check box. This is the case for the `.xls` file you just added, as well as for `.csv` files, and you must deselect this check box prior to setting the FIELD names.

Note – If a data file does not exist, you can select to create one.

g. **In the Add External Tables/Enter Column Properties window, double-click on a column field to edit it.**

The # symbol represents the numerical order of the columns, and in the example `FIELD_1` is the name of the first column. In the example, you are going to change the column names as it makes it easier to know what data is in which column when you merge the spreadsheet and the text file fields together.

Change the column names to names that more appropriately represent the values. Moving forward, change the following spreadsheet column names:

`FIELD_1`	`PRODUCT_IDENTIFIER`
`FIELD_2`	`PRODUCT`
`FIELD_3`	`VENDOR`
`FIELD_4`	`VENDOR_CODE`

- To populate the Preview Table Content columns, click Back once, and then click Next.
- When you are satisfied that the table represented in the Preview Table Content in the lower portion of the Enter Column Properties window is correct, click Finish.

  A message confirms that you created the table successfully. You are now ready to repeat the process and add the text file listed in Before You Begin to the database.

After making your changes, press Enter to see the changes reflected in the Preview Table Content in the lower portion of the Enter Column Properties window.

3 Add external data to the Virtual Database.

Note – As stated in Before You Begin, two data sources are used, a spreadsheet and a delimited text file. You are going to merge them into one table. For this step add the delimited text file to the table, and then fine-tune some of the fields.

---

a. From the NetBeans IDE tool bar select Tools→Virtual Database→Add External Tables.

b. In the Add External Tables window, ensure that the appropriate database from the list of available databases is highlighted before continuing.

In the example, `VirtualMashupDB_1222123456789`, which is the database you created. The underscore–numerical extension has been automatically added to the database name. The extension guarantees that you cannot create two databases with the same name, as no two extensions are the same.

c. Navigate to the directory that contains the delimited text file, select it, and click Add.

The file now appears in two locations under the Choose Data source pane: File name and Selected Table Source. You can remove the file by highlighting the file name in the Selected Table Source and clicking Remove. When you are satisfied that your selection is correct, continue with the procedure.

Note – Although you are selecting a file that is located in a local directory, you can also select data on the web by entering the URL.

---

d. In the Add External Tables/Enter Table Details window, confirm that the table name, encoding, table type, and resource URL are correct.

For a delimited text file, which you are using in this example, the table type should be Delimited Flatfile.

Note – The resource URL in the example is the directory and file name of the delimited text file you selected.

---

e. In the Add External Tables/Import Table MetaData window, set how you want to parse the file.

Notice that your choices have changed considerably from what was available with the spreadsheet.

- Use the drop-down boxes to make your changes.

In the example the pipe represents the delimiter, so change the field delimiter from a comma (,) to a pipe (|).

- If your delimited data does not contain column names as is true for the example, deselect the First line contains field names? check box.
- Review the text file to ensure that you are satisfied with your selections.

Note – If a data file does not exist, you can select to create one.

f. In the Add External Tables/Enter Column Properties window, edit the Column Definitions if desired.

Notice that you now have column names that match the column names in the spreadsheet before you changed them, such as `FIELD_1`, `FIELD_2`, and so on. If you changed the column names for the spreadsheet, and you did, you must also change the column names for the delimited text file. If there are columns that are similar between the two data files, the names you use for the related columns in the delimited data file must match the names you set in the spreadsheet.

Change the column names to names that more appropriately represent the column values. Moving forward, make the following changes to the delimited text file FIELD names:

`FIELD_1`	`VENDOR_CODE`
`FIELD_2`	`VENDOR`
`FIELD_3`	`VENDOR_ADDRESS`

- To populate the Preview Table Content columns, click Back once, and then click Next.
- Click Finish when ready.

  A message confirms that you created the table successfully. You are now ready to verify that your Data Mashup database has the correct tables.

4 Verify that the database has the correct tables in the NetBeans IDE.

a. Under the Services tab expand the Databases node.

b. Right–click the database you created and select Connect.

You are now connected to the database you created.

c. Expand the database, expand the Tables node, right click one of the tables you are using, and select View Data.

Repeat this step to view the other table. The tables used in this procedure are `COMPANY_DATA` and `SUPPLIER_ADDRESS`.

5 **Create a Data Mashup Project Using the NetBeans IDE.**

a. **From the NetBeans IDE tool bar select File→New Project.**

b. **In the New Project window do the following in the Choose Project pane.**

Under Categories	Select SOA
Under Projects	Select Data Mashup Module

c. **In the New Business Process Application window, name your Data Mashup project** `DemoDMProject`**, retain or change the location where you want to create the project, and click Finish.**

The project is created and appears as an active project under the Projects tab.

6 **Create a new EDM.**

a. **Under the NetBeans IDE Projects tab, right–click the** `DemoDMProject` **project you just created, and select New→EDM.**

b. **In the New File window under the Name and Location pane type a name for the Data Mashup file and click Finish.**

For this exercise, name the file `demoDMfile`; the program adds the `.edm` extension.

Note – The `Collaborations` directory has already been created and is in the path, which is listed in the Created File field.

The file name you created under the Data Mashup application package (in the example, `demoDMfile.edm`) appears in the Project tree, the EDM Editor opens in a new blank canvas, and the Tool Operators palette appears.

Next Steps You are now ready to configure your Data Mashup project. For specific instructions on how to do this see "Configuring Data Mashup Projects Using Joins" on page 16.

See Also

- "Configuring and Starting the Sun GlassFish Enterprise Server" on page 7
- "Creating and Deploying a Composite Application to Complete a Data Mashup Project" on page 18
- "Testing the Deployment of a JBI Composite Application" on page 21
- "Viewing the Data" on page 23

For additional information, see:

- Using the Sun Data Mashup Engine

Configuring Data Mashup Projects Using Joins

After creating a Data Mashup project wherein you have brought all your diverse data into a common staging area, you are ready to create a federated view of the data. This topic provides the necessary steps for you to create data joins using the data stored in the staging tables.

Note – Requires Java CAPS 6 Update 1 or Sun GlassFish ESB.

▼ To Add a Join

Before You Begin Before creating a join, you must have a virtual database, Data Mashup project, the NetBeans IDE must be running, and you must be connected to the virtual database.

1 **If necessary, connect to the virtual database.**

  a. **In the NetBeans IDE, select the Services tab and expand the Databases.**

  b. **Right-click the database you want to start and select Connect.**

  In this procedure, start `VirtualMashupDB_1222123456789`.

2 **In the NetBeans IDE Project tree, expand your Data Mashup project.**

Under the default package you should see the Data Mashup file you created when creating a Data Mashup project, `demoDMfile.edm` (see "Designing Data Mashup Projects" on page 9).

Note – `demoDMfile.edm` also appears as a tab in the center pane.

---

3 **In the NetBeans IDE center pane, double-click `demoDMfile.edm` to open the NetBeans IDE Canvas.**

4 **Add the tables you want to join to the canvas.**

  a. **In the EDM Editor tool bar click the Add Table icon.**

Note – If the Projects tab covers some of the tools available for use with the canvas, and you do not see the Add Table icon, click the Projects tab to close the pane or click the Restore Window icon on the canvas tool bar.

---

b. **In the Select Source Table window, select the database.**

In this procedure, use `VirtualMashupDB_1222123456789`. After selecting the database, the tables you added to it appear in the Tables area on the right side of the window. You are using the `SUPPLIER_ADDRESS` and `COMPANY_DATA` tables.

c. **Highlight the tables and click Select.**

Repeat this step if necessary to add both tables to the Table Selection area.

d. **Click OK.**

The window closes and the tables, along with Runtime Input, appear on the canvas. You are now ready to create the join.

5 **From the Table Operators palette, drag the Join operator onto the canvas.**

You are now ready to merge your tables into one table. In the example you are going to merge `SUPPLIER_ADDRESS` and `COMPANY_DATA`.

6 **In the Create New Join View window, click All> to move the tables to the Selected Tables list.**

In the example, both tables appear in the Preview area and are linked to the join. You can edit the join types using the drop-down dialog.

Inner	Returns only the records in the selected tables that match. For this example, retain this default.
Left Outer	Returns all records in the left table regardless if there are any matches with the right table. When there is no matching in the right table, NULL fills in the field.
Right Outer	Returns all records in the right table regardless if there are any matches with the left table. When there is no matching in the left table, NULL fills in the field.
Full Outer	Returns the all records from the left and right tables in the merged table. All fields that do not match are filled with NULL.

7 **Click OK.**

The root join is added to the canvas and is linked to the two tables.

▼ To Edit a Join Condition

Before You Begin Before editing a join, you must have a virtual database, Data Mashup project, the NetBeans IDE must be running, you must be connected to the virtual database, and have added a join.

1 **Right-click the Root Join Widget on the canvas and then click Edit Join Condition.**

The Edit Join Condition window, also called the Condition Builder, opens.

2 Create a join condition in the Edit Join Condition window.

a. **Drag a column from a table onto the empty canvas on the right.**

b. **Drag and drop a comparison, string, or another operator onto the canvas**

In this example, use the equal (=) sign.

c. **Drag a column from the second table onto the canvas to complete the query.**

Even though there are two join conditions that match from the tables, `VENDOR` and `VENDOR_CODE`, you only need to complete one join with the equal sign.

d. **When satisfied with your join conditions, click OK.**

The Edit Join Condition window closes. You are almost ready to create and deploy a composite application which contains your Data Mashup project. This step enables you to view your data on the web. However, first you need to save your project.

3 To save the project, right-click it and select Build Project.

A WSDL is created within the Collaboration node. In the example, `DemoDMProject_demoDMfile_engine.wsdl`.

Next Steps You are now ready to create and deploy a composite application in the Sun GlassFish Enterprise Server. For detailed instructions on how to perform this task, see "Creating and Deploying a Composite Application to Complete a Data Mashup Project" on page 18.

See Also

- "Configuring and Starting the Sun GlassFish Enterprise Server" on page 7
- "Designing Data Mashup Projects" on page 9
- "Testing the Deployment of a JBI Composite Application" on page 21
- "Viewing the Data" on page 23

For additional information, see:

- Using the Sun Data Mashup Engine

Creating and Deploying a Composite Application to Complete a Data Mashup Project

Before a Data Mashup project can be opened and used in a browser, it must be added to a composite application project. You use the composite application project to assemble a deployment unit, which includes the Data Mashup project, for the Java Business Integration

(JBI) server. It is only after a successful deployment of the composite application project that you can open and run your Data Mashup project as a web service in a browser.

Note – Requires Java CAPS 6 Update 1 or Sun GlassFish ESB.

▼ To Create and Deploy a Composite Application

Before You Begin Before you can create and deploy a composite application, you must have created a Data Mashup project, set the join conditions, the NetBeans IDE must be running, and you must be connected to the virtual database. For information on how to perform these tasks, see "Designing Data Mashup Projects" on page 9 and "Configuring Data Mashup Projects Using Joins" on page 16.

Note – To move forward or backward in the procedure, click Next or Back.

---

1 **Create a composite application.**

   a. **In the NetBeans IDE tool bar select File→New Project.**

   b. **In the New Project window, select SOA from Categories and Composite Application from Projects, and then click Next.**

   Note – This step creates an empty Composite Application project that may have multiple BPEL modules and Java Business Integration (JBI) modules. SOA may already be selected.

   ---

2 **Name the composite application project and set the location for it in the New Composite Application Project window, and then click Finish.**

In the example, retain the defaults: `CompositeApp1` (project name) and `C:\Documents and Settings\<username>\My Documents\NetBeansProjects` (project location).

Note – If you do not want this project to be set as a main project make sure you deselect the Set as Main Project check box.

The following happens:

- A Composite Application project is created under the Projects tab in the NetBeans IDE
- The Composite Application Service Assembly (CASA) editor opens up in the NetBeans IDE

You are now ready to add a Data Mashup application to the JBI modules of the composite application.

3 **Under the Projects tab in the NetBeans IDE, add the Data Mashup project and its JAR files to the composite application by right-clicking JBI Modules and selecting Add JBI Module.**

- Right-click the deployment JBI node, in the example, `CompositeApp1`, and select Add JBI Module.
- Right-click JBI Modules and select Add JBI Module.

Either action opens the Select Project window.

4 **In the Select Project window, highlight the Data Mashup application in the list and click Add Project JAR Files.**

In the example, `demoDMProject`. This builds the service engine deployment JAR file. If you do not see the Data Mashup project in the left pane of the window, navigate to the directory that contains it. The Data Mashup application, in the example, `(EDM)DemoDMProject`, is added to the CASA Editor canvas as a JBI module.

5 **In the CASA Editor, right-click in the WSDL Ports area and select Add WSDL Port.**

The Select WSDL Port window appears, listing all the available WSDL ports.

6 **In the Select WSDL Port window, select `http` and `soap` and then click OK.**

An HTTP port named `casaPort1` and a SOAP port named `casaPort2` appear in the WSDL Ports area of the canvas.

Note – Data Mashup only supports the `http` and the `soap` WSDLs.

If you want to use the procedure to test the deployment of your composite application, when creating the WSDL port you must use the `soap` WSDL binding and not the `http` WSDL binding.

7 **Build the composite application.**

Check your output in the bottom pane to ensure that the composite application built successfully.

Note – If the build fails, do the following:

a. In the Projects pane, expand the composite application project, if not already expanded, and select the `*.wsdl` under Process File.

b. In the CASA Editor, select the Source tab.

c. Scroll down until you see

```
<http:address location="http://localhost:${HttpDefaultPort}/
casaservice1/casaPort1"/>
```

d. Change `casaPort1` to another port number, such as `casaPort6`.

e. Build the composite application again.

8 **Drag the purple HTTP and SOAP consumer endpoints in the WSDL Ports pane to the green** `(EDM)DemoDMProject`**.**

9 **Build the composition application a second time.**

10 **Deploy the composite application.**

In this example, you deployed `CompositeApp1`. You should see a message in the Output pane that tells you that the build was successful.

Next Steps If you used the `soap` WSDL binding, you can test the deployment of your composite application. See "Testing the Deployment of a JBI Composite Application" on page 21.

See Also

- "Configuring and Starting the Sun GlassFish Enterprise Server" on page 7
- "Designing Data Mashup Projects" on page 9
- "Configuring Data Mashup Projects Using Joins" on page 16
- "Viewing the Data" on page 23

For additional information, see:

- Using the Sun Data Mashup Engine

Testing the Deployment of a JBI Composite Application

This topic leads you through the steps to test the deployment of your JBI composite application.

Note – Requires Java CAPS 6 Update 1 or Sun GlassFish ESB.

▼ To Test the JBI Application Deployment

Before You Begin Before you can test the deployment of a JBI composite application, you must have created a Data Mashup project, set the join conditions, created and deployed a composite application using the SOAP WSDL port, the NetBeans IDE must be running, and you must be connected to the virtual database. For additional information about these tasks, see:

- "Designing Data Mashup Projects" on page 9
- "Configuring Data Mashup Projects Using Joins" on page 16
- "Creating and Deploying a Composite Application to Complete a Data Mashup Project" on page 18

Note – To move forward or backward in the procedure, click Next or Back.

For this procedure to test the deployment of a composite application to succeed, when creating the WSDL port you must use the `soap` WSDL binding and not the `http` WSDL binding.

1 **In the NetBeans IDE under the Projects tab, right-click the Test node and choose New Test Case.**

2 **In the New Test Case window, enter a name for the test case and click Next.**

For this procedure, use the name `SOAPTestCase1`.

3 **In the New Test Case window, expand the directory structures for the composite application, select the `*.wsdl`, and then click Next.**

For example, `CompositeApp1.wsdl`, which is located under `CompositeApp1 – Process Files`.

4 **In the New Test Case window, select execute for the operation you want to test and click Finish.**

The operation you want to test is `casaPort2 (Binding="casaBinding2")`. You are almost ready to test your deployment. However, before you run the test, you must set the variables in the XML Editor.

5 **In the XML Editor, replace the `?string?` variables with a numeral.**

For example, `0`.

6 **Under the Test node in the Projects pane, right–click the SOAP test case and choose Run.**

In this example, `SOAPTestCase1`.

Next Steps See "Viewing the Data" on page 23 for steps that must be performed before you can view your data on a browser.

See Also

- "Configuring and Starting the Sun GlassFish Enterprise Server" on page 7
- "Designing Data Mashup Projects" on page 9
- "Configuring Data Mashup Projects Using Joins" on page 16
- "Creating and Deploying a Composite Application to Complete a Data Mashup Project" on page 18

For additional information, see:

- Using the Sun Data Mashup Engine

Viewing the Data

Before you can open your Data Mashup virtual database as a web service in a browser, you need to ensure that the URL is set up properly.

There are two tasks you can perform prior to opening your Data Mashup virtual database in a browser.

- Choose which table columns you want to display in the browser
- Set the Data Mashup virtual database so that it functions as desired in the browser

Note – Requires Java CAPS 6 Update 1 or Sun GlassFish ESB.

▼ To Select the Columns That are Exposed in the Browser

1. **In the Projects pane, highlight the Data Mashup project you created.**

   For example, `DemoDMProject`.

2. **In the EDM Editor, choose the Design tab.**

   The ROOT JOIN, Runtime Input, and the `SUPPLIER_ADDRESS` and `COMPANY_DATA` tables appear in the EDM Editor. The columns that will be exposed in the browser are listed in the tables.

   a. **To prevent a column from appearing the browser, right-click a table, such as `SUPPLIER_ADDRESS`.**

   b. **In the selection list that appears in the EDM Editor, choose Select Columns.**

   c. **Deselect the columns you do not want to appear in the browser.**

▼ To Invoke a Web Service with the HTTP Binding Component

1. **Locate and edit the URL for the Data Mashup virtual database.**

   a. **Under the Projects tab, expand the composite application→Process Files, and select the WSDL for the project.**

   For example, `demoCompositeApp` and `demoCompositeApp.wsdl`.

   b. **In the WSDL Editor, choose the Source tab to expose the code for the WSDL.**

   c. **Scroll down until you find the line that is similar to:**

   ```
   <http:address
   location="http://localhost:${HttpDefaultPort}/casaservice1/casaPort1"/>
   ```

   You are going to edit the part of the string that is highlighted in bold. When you are done, the line will look something like: `http:addresslocation="http://localhost:9080/casaservice1/casaPort1`.

   d. **(Optional) If the casaPort numbers are already in use, for example,** `casaPort1` **and** `casaPort2`**, change the number of casaPort in the string to another number, for example** `casaPort6`**.**

   e. **Locate the port number for your localhost.**

      i. **In the Projects pane, expand Servers→GlassFish V2→JBI→Binding Components, and right-click** `sun-http-binding`**.**

      ii. **In the sun-http-binding – Properties window use the default HTTP port number to replace the string,** `${HttpDefaultPort}`**, for example** `9080`**.**

   f. **Type the code that sets how you can display your virtual database in the browser.**

   For example: `?page=0&row=0&column=0&`. When you open your Data Mashup virtual database in the browser, replace the zeros (0) in the string to change how the information for `COMPANY_DATA` and `SUPPLIER_ADDRESS` is displayed.

   g. **Add the two tables you mashed together to the URL.**

      i. **In the Projects pane, highlight the Data Mashup project you created.**

      For example, `DemoDMProject`.

      ii. **In the EDM Editor, choose the Source tab.**

iii. **In the EDM Editor, scroll until you find `FILE_LOC_S1_SUPPLIER_ADDRESS` and `FILE_LOC_S2_COMPANY_DATA`.**

iv. **Add the runtime input, `FILE_LOC_S1_SUPPLIER_ADDRESS` and `FILE_LOC_S2_COMPANY_DATA` with `=0&` after the first file name and `=0` after the second file to your URL.**

If you changed the CASA port number as mentioned above, the URL should now look something like:

```
http://localhost:9080/casaservice1/casaPort6?page=0&row=0&column=0&
FILE_LOC_S1_SUPPLIER_ADDRESS=0&FILE_LOC_S2_COMPANY_DATA=0
```

If you did not change the CASA port number as mentioned above, the URL should now look something like:

```
http://localhost:9080/casaservice1/casaPort1?page=0&row=0&column=0&
FILE_LOC_S1_SUPPLIER_ADDRESS=0&FILE_LOC_S2_COMPANY_DATA=0
```

Note – There are no spaces in this URL.

You are now ready to copy the URL into your browser and view and manipulate your project, which is now functions as a web service.

2 **Open a browser and copy the URL into it.**

3 **Replace the zeros in the URL string to change how the web service displays.**

Note – If you have any trouble viewing the data, see issue #65 in "Known Data Mashup Issues" on page 29.

4 **Repeat step 3 with other values.**

See Also

- "Configuring and Starting the Sun GlassFish Enterprise Server" on page 7
- "Designing Data Mashup Projects" on page 9
- "Configuring Data Mashup Projects Using Joins" on page 16
- "Creating and Deploying a Composite Application to Complete a Data Mashup Project" on page 18
- "Testing the Deployment of a JBI Composite Application" on page 21

For additional information, see:

- Using the Sun Data Mashup Engine

Sun Data Mashup Engine Application Configuration

This topic provides information about the Sun Data Mashup Engine application configuration feature, which enables the dependencies to be externalized and configured. When projects are moved from Development→Staging→Production, this feature can be used to migrate by configuration through the Sun GlassFish Enterprise Server Admin Console or the AppConfig Editor available through the JBI Manager. Using this feature, datasource connectivity configuration at Design–time can be overridden by associating the connection identifiers with `jndi` resources.

Note – Requires Java CAPS 6 Update 1 or Sun GlassFish ESB.

▼ To Configure the Data Mashup Application Engine

1 **To correlate the connections with the Sun Java System Application Server (Sun GlassFish Enterprise Server) Admin Console Configurations screen, see the Edit Database Properties window in the EDM Editor.**

The source connections, for example `SourceConnection1` and `SourceConnection2`, listed in this window are the connections you need to overwrite. Configure the Connection Pools using the Sun Java System Application Server (Sun GlassFish Enterprise Server) Admin Console for the connections identified in step 1.

2 **Start the Sun GlassFish Enterprise Server and open the Admin Console.**

You are going to use Sun Java System Application Server Admin Console window to configure the connection pools you identified in step 1.

3 **In the Admin Console use the Configurations tab to create configuration entries.**

a. **Under Common Tasks, select JBI→Components, and then highlight the component you want to configure.**

For example, `sun-edm-engine`.

b. **Under the Configurations tab, set the** `SourceConnection1` **and** `SourceConnection2`**.**

For example, `SourceConnection1` might equal **`jdbc/derbypool`** and `SourceConnection2` might equal **`jdbc/mysqlpool`**.

4 **Associate the Service Unit Endpoint in the EDM Editor canvas to a particular** `Config Extension`**.**

For example, here you are connecting a port, such as `port1`, to the (EDM) project, such as `(EDM)yourProject`.

5 **Deploy the project.**

After you have deployed the project, your changes to the configuration are applied, providing you with a runtime switch of connections.

See Also

- "Sun Data Mashup Engine Overview" on page 6
- "Configuring and Starting the Sun GlassFish Enterprise Server" on page 7
- "Designing Data Mashup Projects" on page 9
- "Configuring Data Mashup Projects Using Joins" on page 16
- "Creating and Deploying a Composite Application to Complete a Data Mashup Project" on page 18
- "Testing the Deployment of a JBI Composite Application" on page 21
- "Viewing the Data" on page 23
- "Sun Data Mashup Engine REST Support" on page 27
- "Known Data Mashup Issues" on page 29

Sun Data Mashup Engine REST Support

This topic provides information about REST support.

Note – Requires Java CAPS 6 Update 1 or Sun GlassFish ESB.

- Resources
  - Identified Resources
    - Tables, views, federated views, data mashups of relational/rss feed
    - Reaching to particular page, row and column
  - Supported Parameters
    - Page: Supports pagination of large result sets
    - Row: Reaches a particular row irrespective of the page
    - Column: Reaches a particular column
    - Extra parameters: depends upon the application configuration
- Constructing Resource URIs in encoding mode

  Data Mashup Engine support of REST.

  - To get all the records, the URL is:`page=0&row=0&column=0`
  - To reach a particular page, such as the 2nd page, the URL is: `page=2&row=0&column=0`

    If you attempt to reach an invalid page, such as `-1`, the URL returns all the pages.
  - To reach a particular row, such as the 10th row, the URL is: `page=2&row=10&column=0`

The page value is ignored and only 10th row is displayed. If you attempt to reach a row with an invalid row, all the pages and all the rows are returned.

- To get all the records of a particular attribute, such as `CUSTOMER_ID`, the URL is: `page=0&row=0&column=CUSTOMER_ID`

  This lists all the records of attribute `CUSTOMER_ID`.
- To get only values of the `CUSTOMER_ID` in 2nd page, the URL is: `page=2&row=0&column=CUSTOMER_ID`
- To drill down to the 2nd row of the `CUSTOMER_ID` value , the URL is: `page=2&row=2&column=CUSTOMER_ID`

  In this instance, the page attribute value is ignored.

Note – Currently the Data Mashup Engine does not support a POST request.

- Runtime-Input Arguments
  - Runtime-input arguments can be added at Design time
  - Runtime arguments for flat files and tab delimited files are added automatically
- Using Runtime arguments with REST Query
  - The default way to Reach `page=0&row=0&column=0&FILE_LOC_STUDENTINFO=0`

    `FILE_LOC_STUDENTINFO` is the runtime argument. Once the runtime argument is defined at design time, it is mandatory to mention this argument while accessing URL.

    `FILE_LOC_STUDENTINFO=0` is the runtime argument that you are accessing with default values. You can overwrite this value by specifying a file name, such as `page=0&row=0&column=0&FILE_LOC_STUDENTINFO=test.xls`.

Note – `test.xls` is the file name that you overwrite; this must be configured in GlassFish as part of application configuration.

Related Topics

- "Sun Data Mashup Engine Overview" on page 6
- "Configuring and Starting the Sun GlassFish Enterprise Server" on page 7
- "Designing Data Mashup Projects" on page 9
- "Configuring Data Mashup Projects Using Joins" on page 16
- "Creating and Deploying a Composite Application to Complete a Data Mashup Project" on page 18
- "Testing the Deployment of a JBI Composite Application" on page 21
- "Viewing the Data" on page 23
- "Sun Data Mashup Engine Application Configuration" on page 26

- “Known Data Mashup Issues” on page 29

Known Data Mashup Issues

This topic lists known Data Mashup issues.

Issue #37: Derby database failure

The Derby database fails if the application configuration parameter is used; however, the Derby database works fine when the application configuration extension for the project is not used. This issue is connected with the Connection Pool Configuration for Derby. The URL parameter is mandatory and the user has to explicitly add the property. When this is done, Derby work fine without any modifications.

Issue #65: First record missing when viewing data in browser

During runtime, after the pagination is set in the request URL, the retrieval of the first record from the query is missing. You can verify this by executing Show Data in the Join condition in the EDM Editor. What follows are row and column combinations for the of the first row:

- Page, row, and column are mandatory fields (apart from runtime arguments).
- If the `page =0&row=1&column=0`, it should display first page.
- If `page=1&row=1&column=0`, ignore it if it is to show only the 1st row.
- If `page=1&row=-1&column=0` and if it has an invalid row number, it ignores the row and displays the 1st page.
- If `page=1&row=1&column=Field1`, only the value of Field1 of the 1st row is displayed.
- If `page=1&row=1&column=invalid Field` and if the column value is invalid, only the 1st row with all the fields is displayed.

Issue #89: Data Mashup module deploys with an invalid application configuration name

The Data Mashup module is deployed even if the application configuration name that is specified is invalid. For example, if the Application Configuration name changes from `FF_x` to `FF_AC`, this behavior does not prevent the Data Mashup module from deploying.

Note – Requires Java CAPS 6 Update 1 or Sun GlassFish ESB.

See Also

- Using the Sun Data Mashup Engine
- “Sun Data Mashup Engine Overview” on page 6
- “Configuring and Starting the Sun GlassFish Enterprise Server” on page 7
- “Designing Data Mashup Projects” on page 9
- “Configuring Data Mashup Projects Using Joins” on page 16
- “Creating and Deploying a Composite Application to Complete a Data Mashup Project” on page 18

- “Testing the Deployment of a JBI Composite Application” on page 21
- “Viewing the Data” on page 23
- “Sun Data Mashup Engine Application Configuration” on page 26
- “Sun Data Mashup Engine REST Support” on page 27