Using the Sun Data Mashup Engine

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

Part No: 821–0455–10
September 2009

091019@22749

Contents

Using the Enterprise Data Mashup Service Engine

The topics listed here provide instructions on how to use the Enterprise Data Mashup Service Engine (Data Mashup SE), a JBI-compliant service engine (SE) that provides Enterprise Data Federation, or mashup, services. The Data Mashup SE 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.

The Data Mashup SE is automatically installed with a complete Java CAPS 6 Update 1 or GlassFish ESB installation.

What You Need to Know

These topics provide conceptual information.

What You Need to Do

These topics provide instructions on how to use the Data Mashup SE to create, configure, and deploy Data Mashup projects. The topics are provided in the form of an exercise that takes you through the steps of creating a virtual database, pulling in data from multiple sources, and creating a collaboration that maps the data to one target.

Enterprise Data Mashup Service Engine Overview

Use the Enterprise Data Mashup Service 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. The Data Mashup SE is a standalone product. It can, however, expose certain JBI-based MDM Suite data sources as services.

The Data Mashup SE provides server side mashup for dispersed data. More specifically, the Data Mashup SE 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 7
- "Configuring Data Mashup Projects Using Joins" on page 14
- "Creating and Deploying a Composite Application for a Data Mashup Project" on page 17
- "Testing the Deployment of a Composite Application" on page 20
- "Configuring and Viewing the Data" on page 21

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

Starting the GlassFish Server and Data Mashup Service Engine

This topic provides information for starting the GlassFish server and the Enterprise Data Mashup Service Engine. While creating and working on a Data Mashup project in the NetBeans IDE you do not need to have the Sun GlassFish Enterprise Server running. In order to start the service engine and configure its runtime properties, the GlassFish server needs to be started. When Data Mashup requires the application server, Data Mashup starts the server automatically.

▼ To Start the Server and Service Engine

1 **To start the GlassFish server, do the following on the NetBeans IDE:**

   a. **Select the Services tab.**

   b. **Expand the Servers node.**

   c. **Right-click the GlassFish server (GlassFish V2, by default), and then 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 Services panel.

2 **To start the Enterprise Data Mashup Service Engine, do the following:**

   a. **Expand the application server node in the Services panel.**

   b. **Expand JBI.**

   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 instructions on how to do this see "Designing Data Mashup Projects" on page 7.

Designing Data Mashup Projects

The following exercise leads you through the steps of designing a Data Mashup project, and includes the following steps:

- 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

For this exercise you will create a federated join using two data files, a Microsoft Excel spreadsheet (`company-data.xls`) and a delimited text file (`supplier-address.txt`). Ensure that you have the data files accurately set up before beginning the procedure.

Perform the following steps in the order given to develop the Data Mashup project:

- "To Create the External Data Files" on page 8
- "To Create the Data Mashup Database" on page 9

▼ To Create the External Data Files

You can either create the spreadsheet and delimited text file or you can download the files 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` for this exercise.

1 Create an Excel spreadsheet named `company-data.xls` that contains the following information:

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	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
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

2 **Create a delimited text file named `supplier-address.txt` that contains the following information:**

```
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
```

▼ To Create the Data Mashup Database

Before You Begin Before beginning this procedure, you must have the above-mentioned files ready and stored on your computer, and have started the NetBeans IDE.

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

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

For this exercise, name the database `VirtualMashupDB`.

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

4 **Click OK on the configuration message that appears.**

You are now ready to start adding external data to the virtual database you just created.

▼ To Add External Data to the Virtual Database (Excel Spreadsheet)

Before You Begin Make sure you have the two data sources, a spreadsheet and a delimited text file. These will be merged into one table. For this step add the spreadsheet to the table, and then fine-tune some of its fields.

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

2. **In the Choose Virtual Database window, select the appropriate database from the list of available databases.**

   For this exercise, select `VirtualMashupDB`, which is the database you just created.

3. **In the File field on the Choose Data source window, navigate to the directory that contains the spreadsheet and double-click the file.**

   For this exercise, select `company-data.xls`. The file now appears in the Selected Table Source list. You can remove it by highlighting the file name in the Selected Table Source and clicking Remove.

   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.

   ---

4. **Click Next.**

5. **In the Enter Table Details window, confirm that the table name, encoding, table type, and resource URL are correct.**

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

6. **Click Next.**

7. **In the Choose a Sheet window, select the sheet containing the data to use, and then click Preview to view the information.**

   In the example data, the information is in `Sheet1`.

8. **Click Next.**

   Step 5, Choose a (HTML) Table is skipped since no HTML sources were selected. The Import Table MetaData window appears.

9. **In the Import Table MetaData window, define how 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 also the case 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. The available fields on this window vary based on the type of data you are importing into the database.

10 **Click Next.**

11 **In the 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. For this example, change the column names to names that more appropriately represent the values. This makes it easier to know what data is in which column when you merge the spreadsheet and the text file fields together. Use the following column names:

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

12 **To populate the Preview Table Content columns, click Back once, and then click Next.**

13 **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 to the database.

▼ To Add External Data to the Virtual Database (Delimited Text File)

Before You Begin Make sure you have the two data sources, a spreadsheet and a delimited text file. For this step, you will add the delimited text file to the table and then fine-tune some of the fields.

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

2 **In the Add External Tables window, select the appropriate database from the list of available databases.**

In this exercise, `VirtualMashupDB` is the database you created.

3 **Click Next.**

4 **In the File field on the Choose Data Source window, navigate to the directory that contains the delimited text file and then double-click the file name.**

The file now appears in the Selected Table Source list. You can remove the file by highlighting the file name in the Selected Table Source and clicking Remove.

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.

---

5 **Click Next.**

6 **In the 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. The resource URL is the directory and file name of the delimited text file you selected.

7 **Click Next.**

8 **In the Import Table MetaData window, define how to parse the file.**

Note – The available fields on this window vary based on the type of data you are importing into the database.

- 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 in our sample file, deselect the First Line Contains Field Names? check box.
- Review the text file in the Preview panel to ensure the selections are correct.

9 **Click Next.**

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

In this exercise, 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 used in the spreadsheet. Change the column names to names that more appropriately represent the column values. For this exercise, use the following values:

```
FIELD_1    VENDOR_CODE

FIELD_2    VENDOR

FIELD_3    VENDOR_ADDRESS
```

11 **To populate the Preview Table Content columns, click Back once, and then click Next.**

12 **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.

▼ To Verify the Virtual Database Tables

1 **Under the Services tab expand the Databases node.**

2 **Right–click the database you created and select Connect.**

You are now connected to the database you created.

3 **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`.

▼ To Create the Data Mashup Project

Once the virtual database is populated, you can create the Data Mashup project. This project will include an EDM collaboration that defines how the data is mashed up into one table.

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

The New Project Wizard appears.

2 **In the Choose Project window, select SOA under Categories, and select Data Mashup Module under Projects.**

3 **Click Next.**

4 **In the New Business Process Application window, name the Data Mashup project `DemoDMProject` and enter the location where you want to store the project files.**

5 **Click Finish.**

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

▼ To Create the EDM Collaboration

Before You Begin Before beginning this procedure, you must 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 **In the NetBeans IDE Projects panel, right–click the `DemoDMProject` project you just created, and select New→EDM.**

The New File Wizard appears.

2 **On the Name and Location window, 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.

---

3 **Click Finish.**

Note – You could click Next instead to add the virtual database tables to the collaboration. This will be done in a later step for this exercise.

The file name you created, 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 instructions on how to do this see "Configuring Data Mashup Projects Using Joins" on page 14.

Configuring Data Mashup Projects Using Joins

After creating a virtual database, Data Mashup project, and EDM collaboration, you are ready to bring all your diverse data into a common staging area and 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.

Perform the following steps to create and configure a join:

■ "To Complete the Data Mashup Project" on page 17

▼ To Add the Tables to the EDM Collaboration

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, click the Services tab and expand Databases.**

   b. **Right-click the database you want to start and select Connect.**
   In this procedure, start `VirtualMashupDB`.

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

3 **Under Collaborations, double-click the EDM collaboration (`demoDMfile.edm` for this exercise).**
The file opens in the EDM Editor canvas.

4 **Right-click in the EDM Editor canvas and select Add Table.**

5 **In the Select Source Table window, select the virtual database you created earlier (VirtualMashupDB).**
The tables in the database appear under Schema.

6 **Highlight the `SUPPLIER_ADDRESS` and `COMPANY_DATA` tables and click Select.**

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

▼ To Create the Join

This step will merge the `SUPPLIER_ADDRESS` and `COMPANY_DATA` into one table.

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

2 **In the Create New Join View window, click All to move both tables to the Selected Tables list.**
Both tables appear in the Preview area and are linked to the join.

3 **To edit the type of join, click in the field at the top of the join table and select one of the following options:**

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

4 **Click OK.**

The root join is added to the canvas and is linked to the two tables. You can now create join conditions for the tables.

▼ To Create a Join Condition

Once you create a join, so you can edit it as necessary by creating join conditions.

1 **Open the EDM collaboration file in the EDM Editor.**

2 **Right-click the Root Join table on the canvas and then click Edit Join Condition.**

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

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

For this exercise, drag `VENDOR` from the `CUSTOMER_DATA` table.

4 **Drag and drop a comparison, string, or another operator onto the canvas to the right of the column name.**

For this exercise, use the equal (=) sign.

Tip – To add an operator, click the appropriate icon on the toolbar. When the drop-down menu appears, click the icon you want to use and drag it onto the canvas.

---

5 **Drag a column from the second table onto the canvas to the right of the operator.**

For this exercise, drag the `VENDOR` column from the `SUPPLIER_ADDRESS` table.

6 **Click OK.**

The Edit Join Condition window closes.

7 **Click Save All.**

You are now ready to build the Data Mashup project.

▼ To Complete the Data Mashup Project

Once the Data Mashup project is configured and saved, you need to build the project in order to create the WSDL document. This document is automatically generated based on the configuration of the collaboration.

- **To complete the project, right-click the project name and select Build Project.**

  A WSDL document named *ProjectName_CollaborationName*`_engine.wsdl` is created under the Collaboration node . In this exercise, the file is named `DemoDMProject_demoDMfile_engine.wsdl`.

Next Steps You are now ready to create and deploy a composite application in the Sun GlassFish Enterprise Server. For instructions on how to perform this task, see "Creating and Deploying a Composite Application for a Data Mashup Project" on page 17.

Creating and Deploying a Composite Application for 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 (CA) 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. A composite application project can have multiple BPEL modules and Java Business Integration (JBI) modules.

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, allowing you to view your data on the web.

Perform the following steps to create and deploy the Data Mashup composite application:

- "To Create the Composite Application" on page 18
- "To Add the Data Mashup Application to the Composite Application" on page 18
- "To Build and Deploy the Composite Application" on page 20

▼ To Create the 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 7 and "Configuring Data Mashup Projects Using Joins" on page 14.

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

The New Composite Application window appears.

2 **In the Choose Project window, select SOA under Categories and select Composite Application under Projects.**

3 **Click Next.**

The Name and Location window appears.

4 **Enter a name for the project and enter the location for the project files.**

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

---

5 **Click Finish.**

The following happens:

- An empty composite application project is created on the Projects page.
- 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.

▼ To Add the Data Mashup Application to the Composite Application

This step adds the Data Mashup project and its JAR files to the composite application.

1 **On the Projects window, expand the Composite Application Tree, right-click JBI Modules and select Add JBI Module.**

2 **In the Select Project window, select the Data Mashup application in the list (`demoDMProject` for this exercise), and click Add Project JAR Files.**

This adds the Data Mashup application to the CASA Editor canvas as a JBI module and builds the service engine deployment JAR file.

Tip – If you do not see the Data Mashup project in the left pane of the window, navigate to the directory that contains it.

---

3 **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.

4 **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` WSDL bindings. 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.

---

5 **Click Save All.**

6 **In the Projects window, right-click the composite application and click Build.**

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

Note – If the build fails, do the following:

a. In the Projects window, expand the composite application project, and select the WSDL file under Process Files.

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.

---

7 **On the CASA Editor, drag the purple HTTP and SOAP consumer endpoints in the WSDL Ports pane to the green endpoint for the Data Mashup project.**

8 **Click Save All.**

You are now ready to build and deploy the composite application.

▼ To Build and Deploy the Composite Application

1 **In the Projects window, right–click the composite application and click Build.**

A message appears in the Output panel letting you know whether the build was successful.

2 **After the build process completes successfully, right-click the composite application and click Deploy.**

A message appears in the Output panel letting you know whether the deployment was successful.

Note – If the application server was not already started before beginning the deploy process, NetBeans automatically starts the server for you.

Next Steps If you used the `soap` WSDL binding, you can test the deployment of your composite application. For more information, see "Testing the Deployment of a Composite Application" on page 20.

Testing the Deployment of a Composite Application

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

▼ To Test the Composite Application Deployment

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

- "Designing Data Mashup Projects" on page 7
- "Configuring Data Mashup Projects Using Joins" on page 14
- "Creating and Deploying a Composite Application for a Data Mashup Project" on page 17

For this procedure to succeed, you must have used the `soap` WSDL binding and not the `http` WSDL binding when creating the WSDL port.

1. **In the NetBeans IDE Projects window, right-click Test under the composite application you want to test and then click New Test Case.**
   The New Test Case Wizard appears.
2. **Enter a name for the test case and click Next.**
   For this exercise, use the name `SOAPTestCase1`.
3. **In the New Test Case window, expand the directory structure for the composite application process files and then select the WSDL file.**
4. **Click Next.**
5. **In the New Test Case window, select `execute` for the operation you want to test and click Finish.**
   For this exercise, select `casaPort2 (Binding="casaBinding2")`.
6. **Click Finish.**
   A file named `Input.xml` is created and appears in the XML Editor.
7. **In the XML Editor, replace the `?string?` variables with a numeral.**
   For example, `0`.
8. **Under the Test node in the Projects pane, right–click the SOAP test case and choose Run.**
   For this exercise, the SOAP test case is `SOAPTestCase1`.
9. **The first time you run the test, the status of the test is `Failed` and a message appears asking if you want to save the output. Click Yes.**
   Subsequent test runs should pass.
10. **To verify the test output, double-click Output under the test case.**
    The output text appears in XML format in the XML Editor.

Next Steps See "Configuring and Viewing the Data" on page 21 for steps that must be performed before you can view your data on a browser.

Configuring and 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 configured properly.

Perform the following steps to configure and view the mashed up data:

- "To Select the Columns That are Exposed in the Browser" on page 22
- "To Invoke a Web Service with the HTTP Binding Component" on page 22

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.
- Configure the Data Mashup virtual database so that it functions as desired in the browser.

▼ To Select the Columns That are Exposed in the Browser

1. **In the Projects window, expand the Data Mashup project you created and double-click the EDM collaboration.**

   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.

2. **Right-click a table for which you want to select or deselect columns.**

3. **Click Select Columns.**

4. **On the Select Columns dialog box, deselect any columns you do not want to appear in the browser and make sure the columns you want to display are selected.**

▼ To Invoke a Web Service with the HTTP Binding Component

In this step, you need to locate and edit the URL for the Data Mashup virtual database. Locate the information you need to add to the URL by performing the following steps. Make a note of the information you find as you will need to add it to the http:address URL in the WSDL file.

- To locate the default HTTP port number, do the following:
  1. In NetBeans, click the Services tab and expand Servers→GlassFish V2→JBI→Binding Components.
  2. Right-click `sun-http-binding` and click Properties.
  3. In the sun-http-binding – Properties window, note the value of the Default HTTP Port Number property (by default, `9080`).
- Locate the names of the runtime inputs by doing the following:

1. In the Projects window, expand the Data Mashup project you created, and double-click the EDM collaboration (for this exercise, `demoDMfile.edm`).

   The file opens in the EDM Editor.
2. In the EDM Editor, click the Source tab.
3. Scroll to the first `dbTable` element, and look for a runtime input name attribute. Note the value of the attribute.

   For this exercise, the attribute is named `flatFileLocationRuntimeInputName` and the value is `FILE_LOC_S1_SUPPLIER_ADDRESS`.
4. Repeat the above step for the next `dbTable` element.

   For this exercise, the value is `FILE_LOC_S2_COMPANY_DATA`.

To invoke the web service:

1 On the Projects window, expand Process Files under the composite application, and double–click the WSDL file for the project.

The file appears in the WSDL editor.

2 In the WSDL Editor, expand Services and expand casaService1 and casaService2.

3 (Optional) If the casaPort numbers are already in use, for example, `casaPort1` and `casaPort2`, do the following:

a. Select casaPort1, and change the value of the Name property to another port number; for example `casaPort6`.

Tip – If the Properties panel does not appear when you select casaPort1, go to the menu bar and select Windows and then Properties.

---

b. Repeat the above step for casaPort2.

4 Under the first casaPort node, select http:address.

The value of the Location property is similar to `http://localhost:${HttpDefaultPort}/casaservice1/casaPort1`.

5 Modify the Location property URL by changing *${HttpDefaultPort}* to the default HTTP port number you located earlier.

When you are done, the entry will look similar to: `http://localhost:9080/casaservice1/casaPort1`.

6 **Append code to the URL that sets how you can display your virtual database in the browser.**

For example: `http://localhost:9080/casaservice1/casaPort1?page=0&row=0&column=0&`.

When you open the Data Mashup virtual database in the browser, replace the zeros (0) in the string to change how the data is displayed.

7 **Append the runtime inputs to the URL, appending `=0&` to `FILE_LOC_S1_SUPPLIER_ADDRESS` and appending `=0` to `FILE_LOC_S2_COMPANY_DATA`.**

The URL should now look similar to:

```
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.

8 **Click Save All.**

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

10 **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.

Data Mashup Service Engine Application Configurations

This topic provides information about creating an application configuration for the Data Mashup Service Engine. When you design a Data Mashup application, you will likely use development or test data sources rather than production sources. When you move a Data Mashup project to staging and later to production, you might need to change the connection information and other properties for the new environment. Application configurations allow you to do this without having to reconfigure the design-time project.

Application configurations enable these dependencies to be externalized and configured. If you use application configurations to define different data sources, you need to create a connection pool and JDBC resource for each source database. Then you can specify the JNDI name of the resource in the application configuration.

You can create a configuration using either the GlassFish Enterprise Server Admin Console or the Application Configuration Editor, which is accessed through the NetBeans JBI Manager on the Service window. This topic provides instructions for both methods.

To create an application configuration, perform the following steps in the order given:

▼ To View Design-Time Connection Information for Source Tables

You can view a list of source connections and their database properties from the EDM file in a Data Mashup project. This can help you determine whether you need to create an application configuration to define different source databases. an Application configuration can overwrite these values at runtime.

1. **On the NetBeans IDE Project window, expand the Data Mashup project and double-click the EDM file to open it.**
2. **Right-click on the EDM Editor canvas, and then select Database Properties.**
3. **For each source connection listed, select the source connection and view the connection URL and login information.**

▼ To Create Connection Pools and JDBC Resources

For an application configuration, you need to create one connection pool and one JDBC resource for each source database. You can use up to five source connections in an application configuration.

For more information about creating connection pools and JDBC resources, see Chapter 3, "JDBC Resources," in *Sun GlassFish Enterprise Server 2.1 Administration Guide*.

1. **Launch the GlassFish Admin Console.**
2. **For each source database used in the Data Mashup project, create one connection pool.**

Note – The URL property is mandator for Derby connection pools. Make sure to add that property in the connection pool and enter the URL in the correct format.

3 **For each connection pool you created, create a corresponding JDBC resource.**

▼ To Create a Data Mashup Application Configuration (From the Admin Console)

1 **Launch the GlassFish Admin Console.**

2 **Under Common Tasks, expand JBI→Components, and then select sun-edm-engine.**
The Service Engine General Properties page appears.

3 **Click the Application tab, and then click Add Configuration.**

4 **In the Identification section, enter a name for the application configuration.**

5 **In the General Properties section, enter the following information:**

- **DataDir** – The path to the source data files.
- **DynamicFlatFile** – An indicator of whether the Data Mashup SE uses the file name passed in the input query (in the URL or SOAP request, for example). When this property is set to "true", the Data Mashup SE uses the file name in the input query. When this property is set to "false", the Data Mashup SE constructs the file path using the value of the `DataDir` property plus the file name passed in. When this property is not set, it defaults to `user.home`.
- **SourceConnection#**– The JNDI names of the JDBC resources you created above.
- **WorkingDir** – The directory where temporary virtual database instances are created. If not set, this defaults to `user.home`.

6 **Click Save.**

▼ To Create a Data Mashup Application Configuration (From NetBeans)

1 **On the NetBeans IDE, click the Services tab.**

2 **Start the GlassFish server.**

3. **Expand Servers > GlassFish V2 > JIB > Service Engines.**
4. **Right–click sun-edm-engine and select Properties.**
5. **Click the Ellipses button next to the Application Configuration property.**
   The Application Configuration Editor appears.
6. **Click Add.**
   An empty row appears.
7. **Enter the following information (click in the blank cell under a column name to add a value):**
   - **Configuration** – A unique name for the application configuration.
   - **DataDir** – The path to the source data files.
   - **DynamicFlatFile** – An indicator of whether the Data Mashup SE uses the file name passed in the input query (in the URL or SOAP request, for example). When the property is set to "true", the Data Mashup SE uses the file name in the input query. When the property is set to "false", the Data Mashup SE constructs the file path using the value of the `DataDir` property plus the file name passed in. When this property is not set, it defaults to `user.home`.
   - **SourceConnection#**– The JNDI names of the JDBC resources you created above.
   - **WorkingDir** – The directory where temporary virtual database instances are created. If not set, this defaults to `user.home`.
8. **Click Save.**

▼ To Associate the Application Configuration With an Endpoint

Once you create the application configuration, you need to associate it with a composite application service unit endpoint. Then you can deploy the application and the new configuration is applied.

1. **In NetBeans, open the Service Assembly file that contains the Data Mashup project to configure. If the Service Assembly file does not exist, do the following:**
   - Create the composite application project.
   - Add the Data Mashup project to the composite application.
   - Build the composite application project.
   - Expand the composite application node, and then double-click on the Service Assembly node to open the CASA Editor.

2. **On the CASA Editor, right-click the endpoint with which you want to associate the application configuration, and then click Properties.**

3. **In the Config Extension section of the Properties window, enter the name of the application configuration you created earlier.**

4. **Save the Service Assembly.**

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.

Data Mashup Service Engine REST Support

This topic provides information about REST support.

- Resources
  - Identified Resources
    - Tables, views, federated views, data mashups of relational or RSS feeds
    - 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 depend upon the application configuration
- Constructing Resource URIs in encoding mode

  Data Mashup Service 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 Service Engine does not support a POST request.

- Runtime input arguments
  - Runtime-input arguments can be added in the EDM collaboration at design-time.
  - Runtime arguments for flat files and tab delimited files are added automatically.
- Using runtime arguments with REST queries.
  - 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 in the collaboration, 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.

Known Data Mashup Issues

This topic lists known Data Mashup issues.

https://open-dm-ds.dev.java.net/issues/show_bug.cgi?id=89https://open-dm-ds.dev.java.net/issues/sh – Data Mashup module deploys with an invalid application configuration name

Issue 37 (`https://open-dm-ds.dev.java.net/issues/show_bug.cgi?id=37`) – Derby database failure

The Derby database fails if an application configuration is used and the URL is not explicitly defined; however, the Derby database works fine when the application configuration for the project is not used. This issue is connected with the Connection Pool configuration for Derby. The URL property is mandatory and you must explicitly add the property.

Issue 65 (`https://open-dm-ds.dev.java.net/issues/show_bug.cgi?id=65`) – The first record is missing when viewing data in a browser

During runtime, after the pagination is set in the request URL, the retrieval of the first record from the query. 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 `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 (`https://open-dm-ds.dev.java.net/issues/show_bug.cgi?id=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 does not prevent the Data Mashup module from deploying.