# M3 Web Services Designer User Guide

Version 10.3.x
Published May 2013

### Important Notices

The material contained in this publication (including any supplementary information) constitutes and contains confidential and proprietary information of Infor.

By gaining access to the attached, you acknowledge and agree that the material (including any modification, translation or adaptation of the material) and all copyright, trade secrets and all other right, title and interest therein, are the sole property of Infor and that you shall not gain right, title or interest in the material (including any modification, translation or adaptation of the material) by virtue of your review thereof other than the non-exclusive right to use the material solely in connection with and the furtherance of your license and use of software made available to your company from Infor pursuant to a separate agreement, the terms of which separate agreement shall govern your use of this material and all supplemental related materials ("Purpose").

In addition, by accessing the enclosed material, you acknowledge and agree that you are required to maintain such material in strict confidence and that your use of such material is limited to the Purpose described above. Although Infor has taken due care to ensure that the material included in this publication is accurate and complete, Infor cannot warrant that the information contained in this publication is complete, does not contain typographical or other errors, or will meet your specific requirements. As such, Infor does not assume and hereby disclaims all liability, consequential or otherwise, for any loss or damage to any person or entity which is caused by or relates to errors or omissions in this publication (including any supplementary information), whether such errors or omissions result from negligence, accident or any other cause.

Without limitation, U.S. export control laws and other applicable export and import laws govern your use of this material and you will neither export or re-export, directly or indirectly, this material nor any related materials or supplemental information in violation of such laws, or use such materials for any purpose prohibited by such laws.

### Trademark Acknowledgements



### Publication Information

Release: 10.3.x

Publication date: May 3, 2013

Document Number: MWSDUG_10.3.x_W_01

# Version Log

The version log describes the changes between versions of this document.

| Part Number | Release Date | Description |
|---|---|---|
| MWSUG-910W-01 | December 2009 | First release of 9.1 version |
| MWSDUG-91W-01 | May 2010 | Compatibility with Eclipse 3.5.<br>New icons for web services<br>Ability to override data on test runs<br>New information on using X.509 policy. |
| MWSDUG-92W-01 | May 2011 | First release of 9.2 version. Discontinued support for interactive web services. New Grid Management Pages. |
| MWSDUG-103W-01 | May 2013 | First release of 10.3 version, rebranded for M3. |

# Contents

# Introduction

1



## M3 Web Services Designer Overview

M3 Web Services Designer is a tool that provides the capability to design, model, deploy, and test web services. These web services wrap the business functionality provided by supported back-end applications.

The intended audience for this document are business and technical consultants who are familiar with:

- Web services concepts
- Back-end application providing the business functionality
- M3 application functionality knowledge as appropriate for the services being created.

The current version of M3 Web Services Designer supports the creation of web services for SQL queries and M3 applications, using web services standards. M3 Web Services Designer is designed as an Eclipse plug-in.

M3 Web Services Designer also provides the following functionality:

- Reuse Movex Web Services Framework (MWSF) 1.6 in M3 Web Services Designer by migrating the 1.6 services without having to recreate them
- Export and import of metadata to and from the M3 Web Services Designer.

# Getting Started With M3 Web Services Designer

2

This section describes the procedures involved in installing the current version of M3 Web Services Designer. It also describes the steps necessary to start working with the application.



## Installing M3 Web Services Designer

**Before you start** Note the following guidelines:

- You can have multiple instances and versions of the M3 Web Services Designer installed on your desktop, but you must install a separate Eclipse package for each instance of the Web Services Designer.
- Each Web Services Designer installation must have access to a Web Services server instance. The server and Designer versions must match.

### Download prerequisites

1. Download and install Eclipse 3.7 from http://www.eclipse.org/
2. Download M3 Web Services Designer from the product download record on Infor Xtreme.
3. Put the installation file in a directory on your local drive.

   **Example**

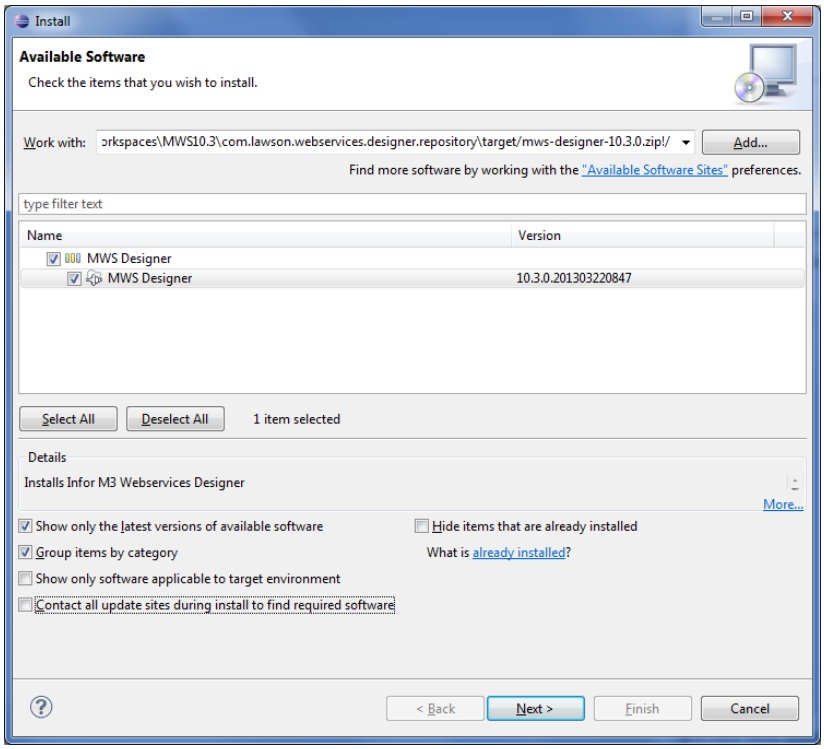
   C:\MWS

### Install M3 Web Services Designer

1 Start Eclipse.

2 In the Eclipse menu, select Help >Install New Software. Select the Available Software tab.

3 Click Archive... Browse to locate the Designer in the directory where you downloaded the installation file.




4 Select the check box next to the MWS Designer and click Next. Install the Designer.

After successful installation, a message is displayed prompting you to restart the workbench for the changes to take effect.

5 Restart the workbench.

## Verifying the M3 Web Services Designer Installation

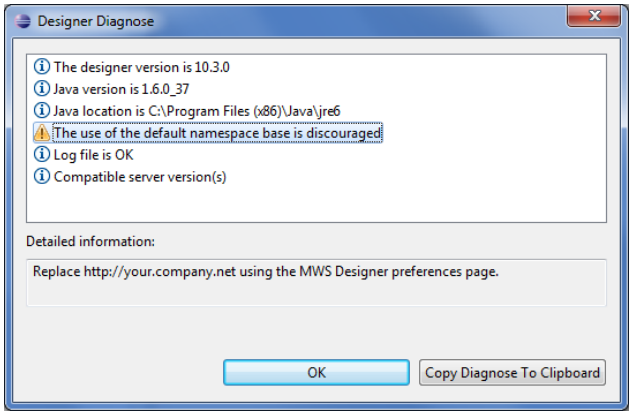
Use the Designer Diagnose tool to verify that the Designer is successfully installed.

1 Open Eclipse.

2 Select MWS Designer > Designer Diagnose.

3 Select Yes to include all servers and click OK.

**Note:** The namespace message is normal. When you configure the M3 Web Services Designer developer options, you will change the default namespace. For more information, see "Specifying Development Settings" on page 17.




# Opening the M3 Web Services Designer Perspective

When you start Eclipse for the first time, the default Resource perspective is displayed. To start working with M3 Web Services Designer, you must change to the M3 Web Services Designer perspective, which provides the Designer views within the workbench window.

1. Launch Eclipse.
2. Navigate to Window > Open Perspective > Other.
3. Select MWS Designer, and then click OK.

   The Web Service Repository and Server tabs appear in the left pane.

# Configuring M3 Web Services Designer

3

This section describes how to set up the M3 Web Services Designer.

Before you begin creating web services with M3 Web Services Designer, you must configure the tool



## M3 Web Services Designer Initial Setup: Process Overview

| Step | Task | For more information |
|---|---|---|
| 1 | Create a repository for web services. | "Web Services Repositories Overview" on page 13"Creating Web Service Repositories" on page 14 |
| 2 | Add a M3 Web Services server. | "Adding an M3 Web Services Server" on page 16 |
| 3 | Configure development preferences. | "Specifying Development Settings" on page 17 |

# Web Services Repositories Overview

Before you start creating web services using the M3 Web Services Designer, you must create a web service repository. The web service repository stores metadata for the web services that are created. Metadata is the information about a back-end system services required in order to generate the corresponding web services.

Repositories are used to store the M3 Web Services web services. A local repository uses a directory on a local hard disk (or file server) to store the M3 Web Services metadata. Local repositories can be shared between Designer users by creating the repository on a shared location.

See "Creating Web Service Repositories" on page 14

**Important:** Ensure that no .XML files apart from web services are stored in the same directory as your local repository. This may cause problems with the repository unintentionally showing external files as web services.

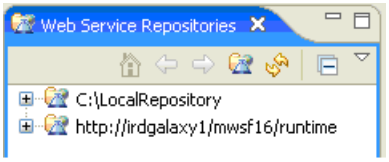
Web service repositories are displayed on the Web Service Repository tab in the Designer. The following options are available from the toolbar.

| Name | Description |
|---|---|
| Home | Displays the hierarchy of the selected Web Service Repositories |
| Back | Returns to the previous view of the Web Service Repositories view |
| Go Into | Displays the web services available for a selected Web Services Repository |
| Add Repository Location | Opens a new web service repository location window |
| Refresh | Refreshes the contents of the Web Service Repositories view |
| Collapse All | Collapses all expanded Web Service Repository contents in the view |
| Menu | Contains menu items that allow you to sort or filter the contents of the Web Service Repositories view |
| Minimize | Minimizes the Web Service Repositories view |
| Maximize | Maximizes the Web Service Repositories view |

# Creating Web Service Repositories

The Web Service Repositories view displays information about the contents of the web service repositories and the way resources relate to each other in a hierarchical manner, which is represented as a tree structure.

Each of the web services and methods are associated with an editor. Editors are views that are designed to view and edit data. Tabs in the editor area indicate the names of resources that are currently open for editing. To work on any resource or see the details in the editor, double-click on the resource in the Web Service Repositories view.

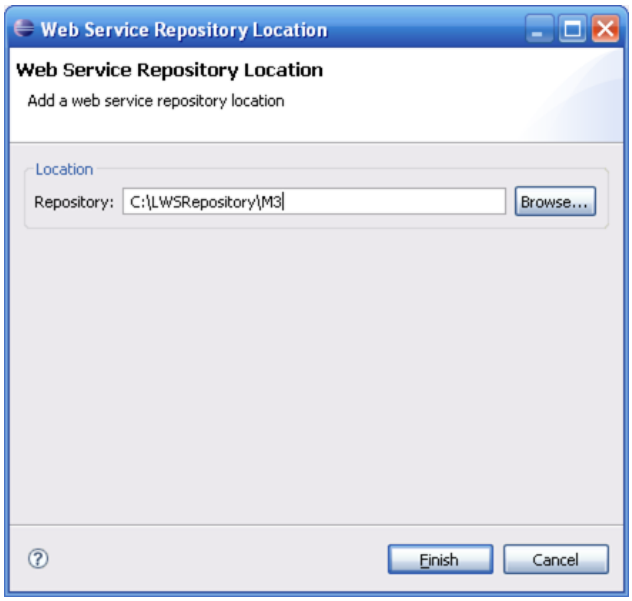



### To create a web service repository

1 Follow any one of the methods to start the Web Service Repository wizard:

- Navigate to File > New > Repository Location.
- Click the Add repository location icon in the toolbar.

The Web Service Repository Location window is displayed.




2 Choose the location where you want to store the repository by using the Browse button. You can either choose an existing folder or create a new folder.

3 Click Finish.

The new repository is created and is displayed in the Web Service Repositories view.

# Changing the Repository Name

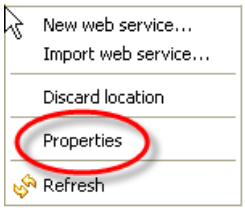
By default, the repository name indicates the path of the web service repository.

For example, while creating the new repository, if you have specified **`U:\MWS_Repositories`** in the Folder field and **`PurchaseServices`** in the Name field, then the name of the repository is **`U:\MWS_Repositories\PurchaseServices`**.

Use this procedure to customize the name of the repository to something that relates the name with the work phase such as the development phase or production phase and accordingly have names such as Development, Production, etc.

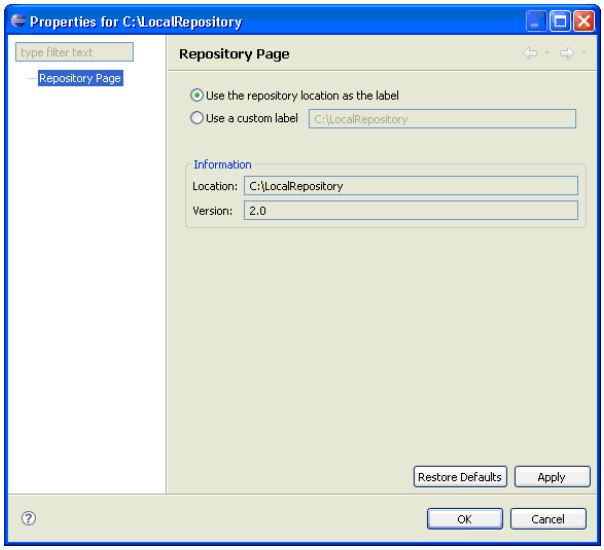
1 In the Web Service Repositories view, right-click the repository whose name you want to customize.

2 Select Properties.




The Repository window is displayed.

**Note:** The screenshot below is for Local repository.




3 Choose Use the repository location as the label if you want to keep the location as the repository name. This is the default option.

-or-

If you want to provide a custom name, choose Use a custom label.

**Tip:** Use a unique name. You can rename the repositories as Development, Production, and Test.

4 Click OK.

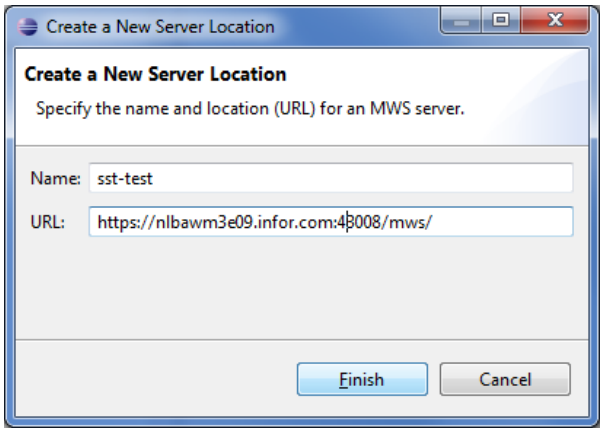
## Adding an M3 Web Services Server

The M3 Web Services Server location provides the path to an M3 Web Services Server. One or more M3 Web Services Server locations must be defined for the M3 Web Services Designer to operate. Server locations are displayed and maintained on the Server tab in the M3 Web Services Designer.

**Note:** When you access the Server for the first time, you will be asked to provide the server user name and password. The credentials will be used when accessing the M3 back-end systems tied to the M3 Web Services Server. Login information will be persisted during your Designer session, and you will not need to log in again. If you want to reset the cached user information and log in as a different user, go to Window > Preferences > MWS Designer > Communication Settings and click Clear Cached Password. For more information, see "Specifying Communication Settings" on page 63.

1 In the Server tab, click +

The Create a new Server Location window is displayed.




2 In the Name field, specify the name of the M3 Web Services Server. This can be a name of your choice.

3 Provide the M3 Web Services Server location in the URL field.

The URL format should follow the **`http://`***`Host:Port/Context_Root`* format.

For example, **`http://seliw136:9080/MWS`**

4 Click Finish.

The M3 Web Services Designer contacts the M3 Web Services Server.

- If the M3 Web Services Server is online, the M3 Web Services Server location is added to the list of servers.

- If the M3 Web Services Server is offline, a message displays:

  **`The selected server seems to be offline, add it anyway?`**

  Click OK if you want to add the M3 Web Services Server location.

  -or-

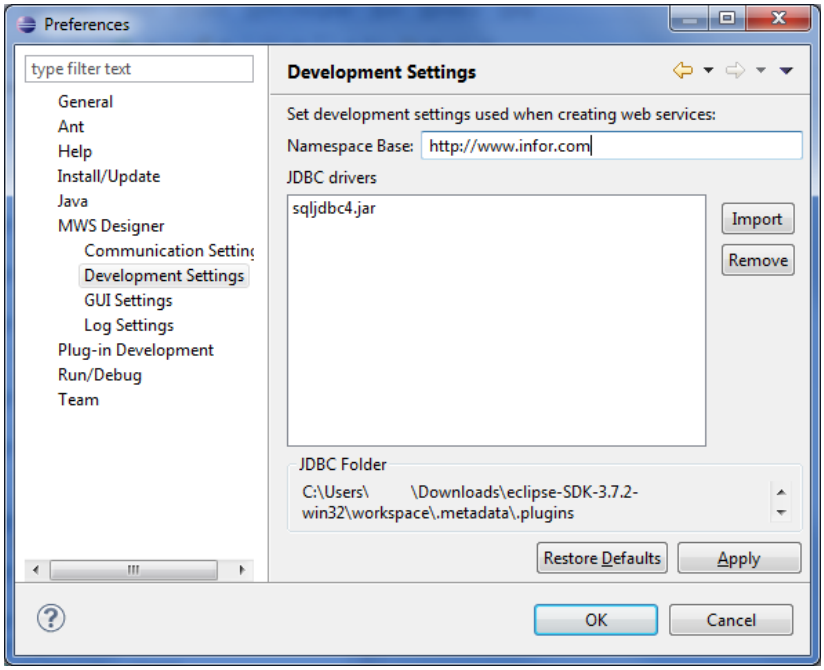
  Click Cancel if you want do not want to add this server location.

## Specifying Development Settings

Use this procedure to specify namespace base, Java compiler, and JDBC drivers.

**1** From Eclipse, choose Window>Preferences.

**2** Expand MWS Designer and choose Development Settings.




**3** Complete the following fields:

| | |
|---|---|
| **Namespace Base** | Namespace acts as a unique qualifier for identifying a web service. The namespace in this context is made up of the Namespace Base followed by the web service name.<br><br>For example, the namespace value of **`http://schemas.M3.com/OrderEntry`** means that **`http://schemas.M3.com/`** is the Namespace Base because it is a valid URI that uniquely identifies the company to which the web service belongs once the web service is published. The **`OrderEntry`** in the URI is the web service name that is appended to the Namespace Base.<br><br>The Namespace Base field contains a default value of **`http://your.company.net`**. Change this value to something to suit your requirement.<br><br>**Note:** For more information about valid URI, go to **`http://www.w3.org/Addressing`**. |
| **JDBC Drivers** | When using the Populate Output from Statement feature in SQL web services, JDBC drivers are needed to be able to call the database. These drivers can be added either by using the Import button. See"Populating Output" on page 48.<br><br>If you are going to work with SQL Queries, import the appropriate JDBC driver by clicking Import.<br><br>The appropriate JDBC driver must also be added on the M3 Web Services Server side. Refer to the *M3 Web Services Administration Guide* for detailed instructions. |

**4** Click OK.

## Specifying GUI Settings

The GUI Settings are used to specify the settings for the M3 Web Services Designer user interface.

**1** From Eclipse, choose Window>Preferences.

**2** Expand MWS Designer and choose GUI Settings.

**3** Set the length of the lists to display on your screen (number of rows). The default is 10.

## Specify Log File Settings

**1** From Eclipse, choose Window>Preferences.

2 Expand MWS Designer and choose Log Settings.

3 Choose the logging option from the drop-down list.

| Option | Description |
|---|---|
| WARN | Displays both warnings and errors. This is the recommended setting for normal Designer use |
| DEBUG | If you are experiencing problems, you can set the log level to DEBUG for more detailed logs |
| ERROR | Displays normal status messages and errors only. Warnings are not displayed. |

## Removing a Web Service Repository

Use this procedure to remove a web service repository from the Web Services Repositories view.

When you remove a web service repository, the repository is not removed from the file system. You can recover the repository by re-creating it in the M3 Web Services Designer and using the location on the file system that already exists. The web services that have been deployed to the server will still be available for consumption.

To delete the web services themselves, see "Undeploying a Web Service" on page 32.

1 In the Web Service Repositories view, right-click the web service repository that you want to discard.

2 Select Discard Location to delete the web service repository.

A message is displayed prompting you to confirm the deletion.

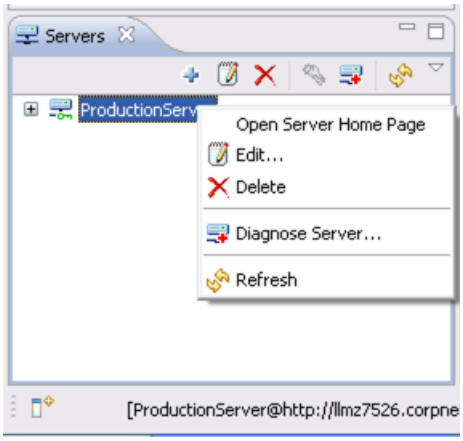
3 Click Yes.

The web service repository is removed from the Web Service Repositories view. However, it is not removed from the file system.

## Removing an M3 Web Services Server Location

Use this procedure to remove an M3 Web Services Server location.

1 On the Server tab, right-click on the server name and choose Delete.




**2** Click OK to delete the server location.

The M3 Web Services Server location is deleted and the entry is removed from the server list.

# Working with Web Services

4

This section describes how you work with web services.



## Creating and Deploying Web Services: Process Overview

| Step | Task | For more information |
|---|---|---|
| 1 | Create a web service. | "Creating a Web Service" on page 22 |

| Step | Task | For more information |
|---|---|---|
| 2 | Create a method. | Follow the instructions for the method you want to use:<br>• "Working with M3 Display Programs" on page 36<br>• "Working with M3 API programs" on page 41<br>• "Working with SQL Queries" on page 44 |
| 3 | Deploy the web service. | "Deploying a Web Service" on page 23 |
| 4 | Test the web service. | "Testing a Web Service" on page 25 |

## Creating a Web Service

The Web Service wizard is used to create the metadata that describes a web service. The actual web service is created only when you deploy to an M3 Web Services Server.

If you create a web service without configuring the M3 Web Services Server, the default namespace (http://your.company.net) is used.

To change the namespace that reflects your company domain, you need to edit the namespace.

**1** In the Web Repository view, right-click the repository where you want to add the new service and choose New.

The Create a new Web Service window is displayed.

**2** Provide a name for the new web service in the Name field.

The name should conform to the XML naming conventions. For information about XML naming conventions, refer to **`http://www.w3.org/TR/REC-xml/`**.

**3** Click Finish.

**Note:** If the checkbox Start the new… is checked, the wizard for creating a new method is started. For more information, see "Working with Web Service Methods" on page 34.

The metadata of the Web Service is created, and is listed in the Web Service Repositories view.

# Changing a Web Service Namespace

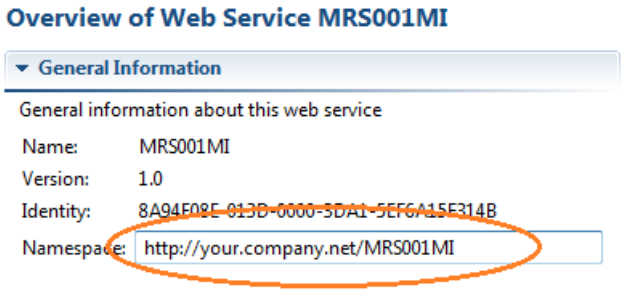
Use this procedure to edit the namespace of a specific web service. You can edit the namespace if you want to change the default namespace to something that is relevant to your company name.

**Important:** Make sure that the changed namespace is a valid URI.

Deploy the web service for the changes to take effect.

1. Go to the Web Service Repositories view and double-click on the web service whose namespace you want to edit.

   The details of the web service are displayed in the editor on the right-hand side.




2. Edit the namespace in the Namespace field.

   **Important:** Make sure that you provide a valid URI.

   For information about URIs, go to http://www.w3.org/Addressing.

3. Select File > Save or use the shortcut keys (Ctrl + S) to save the changes.
4. Deploy the web service after editing the Namespace.

# Deploying a Web Service

Once you have created a web service, you are ready to make that web service available for consumption. The web service is consumed by customers, business partners or other software components within your organization. The process of making a web service available on a production server and publishing its location is known as deployment.

Before deployment, only the metadata for the web service exists. During deployment, the web service is generated and then deployed to the M3 Web Services Server.

**Before you start** Before you deploy the web service, ensure that the following criteria are met:

- Your configuration must be tied to one or more service contexts See "Managing Service Contexts" in the *M3 Web Services Administration Guide.*
- A web service needs at least one method before it can be deployed. For information about creating methods, see "Working with Web Service Methods" on page 34.
- Check the web service and method icons in the repository view.
  - A green icon means that there are no errors or warnings.
  - A yellow triangle indicates that there are warnings, but the web service can be deployed.
  - A red square indicates that there are errors and the web service is not deployable.

## To deploy a web service using the drag-and-drop method

1. In the Web Service Repositories view, expand the repository to locate the web service you want to deploy.
2. Select the web service. Drag it to the Server where you want to deploy it.

   The Select Service Context window is displayed.
3. From the drop-down list, choose a service context. The drop-down lists the options are compatible with the methods defined on the web service.
4. Click Finish.

   If there is already a web service with the same name on the selected server, a warning appears.
   - To stop deploying a particular web service, click No.
   - To cancel the deployment, click Cancel.
   - To replace the existing web service (or web services), click Yes or Yes to All.

   The web service is deployed to the M3 Web Services Server and a message is displayed stating that the web service has been deployed.

## To deploy a web service using the Deployment Wizard

1. In the Web Service Repositories view, expand the repository to locate the web service you want to deploy.
2. Right-click on the web service and choose Deploy.

**Note:** If you didn't specify server settings or forgot to save the web service, you will be prompted for one or more of the following actions before you can deploy:

- Add a server location
- Save the modified resources

For information about how to specify server settings, see"Configuring M3 Web Services Designer" on page 12.

3 From the Select server window, choose the server to deploy to.

For information about adding server locations, see "Configuring M3 Web Services Designer" on page 12.

4 From the drop-down list, choose a service context. The drop-down lists the options are compatible with the methods defined on the web service.

5 Enter the appropriate user name and password for the back-end system

6 Click Finish.

If there is already a web service with the same name on the selected server, a warning appears.

- To stop deploying a particular web service, click No.
- To cancel the deployment, click Cancel.
- To replace the existing web service (or web services), click Yes or Yes to All.

The web service is deployed to the M3 Web Services Server and a message is displayed stating that the web service has been deployed.

After the web service has been deployed, you should test it.For more information, see "Testing a Web Service" on page 25.

## Testing a Web Service

Use this procedure to test a web service. It is good practice to do this when the logic is implemented. M3 Web Services Designer provides the means to create test cases and run them to validate the web service. You can also test a web service using incorrect input data. This allows you to check the error messages that should be displayed for incorrect input data as well as interactivity for these types of programs.

You can reach the test functions by selecting the Test tab in the editor view of a method or by clicking ‘Test tab’ link in the Overview tab.

**Important:** You cannot use the Web Services Designer Test feature to test secured web services. If you implemented security, you must use a third party application to test.

For information about methods, see "Working with M3 Display Programs" on page 36,"Working with M3 API programs" on page 41, and "Working with SQL Queries" on page 44.

Testing a web service involves two steps:

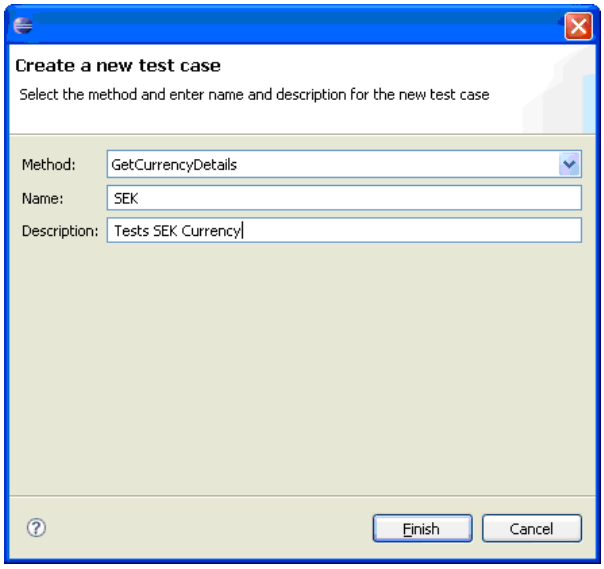
1 Creating a test case

2 Running the test case

## To create a test case

1 In the Test tab link of the Editor view click New to create a test case.

The Create a new test case window is displayed.




2 Select the web service method that you want to test from the Method drop-down list.

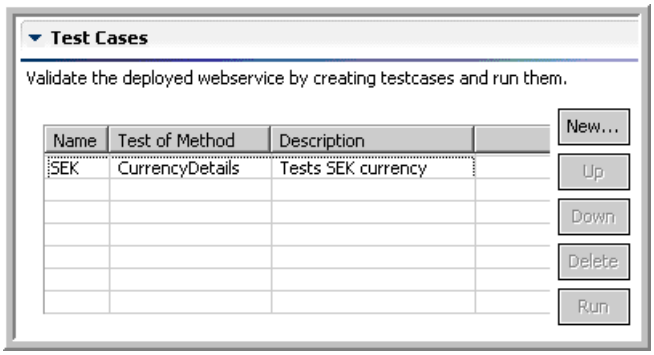
3 Provide a name for the test case in the Name field.

**Tip:** It is good practice to provide meaningful names that describe the input data. If you, for example, are testing currencies you might have a test case called SEK for Swedish Kronor and another called USD to test US Dollars.

4 Enter a description for the test case in the Description field and click Finish.

The test case is now displayed in a test case table.

**Note:** You can modify the description directly from the test case table.



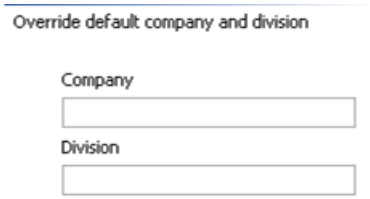

## To run a test case

1 Select a test case that you have created.

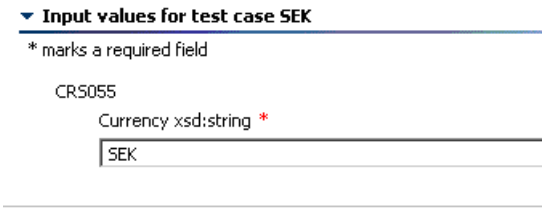
In the example included here, the test case is created for a method wrapping an M3 Display Program, CRS055. The only field that requires an input is WQCUCD (Currency). For more information about M3 Display Programs, see "Working with M3 Display Programs" on page 36.

2 Provide the input values for the test.

- **M3 Programs only** For M3 Display Programs, you can override Company and Division data. For MI programs, you can override Company, Division, and Max Records. If you have saved override values for previous test runs, these fields will be visible when you open the test tab. To save overrides, press **Ctrl-S**.



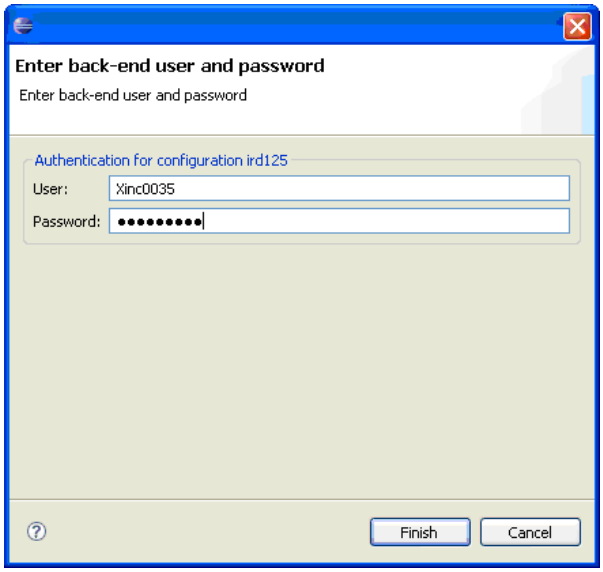

- If the web service method under test contains fields marked as mandatory in its editor, values for those fields are required when the test case is run. Mandatory fields are marked with an asterisk. Optional fields are not required, but ensure that the method's input fields are compatible with the back-end system before you test.



3 You can add expected output values, which are checked against the results of the execution. After the execution, the application returns a result, which may either be **OK** or **FAILED**. Adding expected output values is most useful when running multiple regression tests where the result is known. For more information, see "Testing Multiple Web Services" on page 31.

4 Click Run.

The Enter back-end user and password window is displayed.




5 Provide the back-end server user and password. Click Finish.

**For an M3 API program:**

The results are displayed in the Test Results section. You can use the MI Test tools to verify the output.

**For an M3 display program:**

- The results are displayed in the Test Results section. Run the program and note the output. The test result should display the same output.

# Renaming a Web Service

The M3 Web Services Designer allows you to change the name of an existing web service. You can use the Rename option if, for example, you want to give your web service a more meaningful or descriptive name that the web service consumer will understand.

**Note:** You must redeploy the web service after renaming it. The web service with the old name will continue to run on the M3 Web Services server, so you must undeploy the web service with the old name.

1. In the Web Service Repositories view, select the web service that you want to rename and right-click.
2. Select Rename to rename the web service.

   The Rename window is displayed. The window displays the current web service name.
3. Type the new name for the web service in the New name field.

   Provide a meaningful name for the web service.
4. Click OK.

   The web service is renamed.
5. Deploy the renamed web service. See "Deploying a Web Service" on page 23
6. Undeploy the web service with the old name. See "Undeploying a Web Service" on page 32.

## Copying a Web Service

Use this procedure to copy a web service. This feature is used to copy a web service from one web service repository to another in the M3 Web Services Designer.

1. In the Web Service Repositories view, expand the repository where the web service that you want to copy is located.
2. Select the web service that you want to copy and right-click.
3. Select Copy to copy the web service.

   The selected web service is copied to the clipboard.
4. In the Web Service Repositories view, select the repository where you want to copy the web service to and right-click.
5. Select Paste.

   The web service is now copied to the new web service repository location. All the methods belonging to this web service are also copied.

## Exporting a Web Service

Use this procedure to export web service metadata as an XML file in order to share it with other consultants, or to send it to support personnel. When you export a web service as an XML file, only the metadata of the web service is exported from the M3 Web Services Designer. If the receiver wants to use this metadata as a web service, then the XML file has to be imported into the M3 Web Services Designer (at the receiver's side) and deployed to the M3 Web Services Server.

**Important:** Ensure that the same version of the M3 Web Services Designer is being used when exporting and importing because of possible compatibility issues.

1 In the Web Service Repository, right-click on the web service that you want to export.

2 Select Export.

The Browse for Folder window is displayed.

3 Select a folder or create a new folder using the Make New Folder button.

4 Click OK.

The metadata of the web service is exported as an XML file to the selected folder.

# Importing a Web Service

Use this procedure to import the metadata for a web service into the repository. To create and use the web service, it must then be deployed to a M3 Web Services Server.

**Important:** Ensure that the same version of the application is being used when importing and exporting because of possible compatibility issues.

There are two ways of importing web services:

- From a metadata XML file in the file system
- Directly from a deployed Web Service.

### To import from a metadata XML file in the file system

1 In the Web Service Repositories view, expand the repository where you want to import the XML file of the web service and right-click.

2 Select Import web service.

The Select a web service to import window is displayed.

3 Navigate to the folder where the XML file is stored, select the XML file and click Open.

The XML file is imported into the selected web service repository.

### To import directly from a deployed web service

1 From the M3 Web Services Server Home Page, select List Web Services.

A list of deployed web services is displayed.

2 Select the service you want to import.

The service information is displayed.

3 Select the Meta Data link.

The service metadata is displayed.

4 In the Web Service Repositories view, expand the repository where you want to import the XML file of the web service and right-click.

5 Select Import web service.

The Select a web service to import window is displayed.

6 Copy the metadata URL and paste this into the File field in the Import window. Click Finish.

The service is imported into the repository.

## Copying a Web Service's Endpoint

It is sometimes necessary to know the URL for a web service. You can copy a web service's endpoint to your clipboard, and paste it into another document or tool.

1 In the Server tab, click to expand the server and locate the web service you want to view.

2 Right-click on the web service and choose Copy Endpoint.

3 In the location where you want to place the URL, press Ctrl-V to paste the contents of the clipboard.

## Testing Multiple Web Services

It is possible to test multiple web services that has one or more test cases. This procedure is particularly useful when combined with the expected value for a test case.

The result of the test is given in a test report at specified location.

1 On the Navigator view, select the web services that you want to test.

**Important:** You should only select web services that are deployed to the same service context.

2 Right-click the selected web services, and then select Run test(s).

3 Specify an output directory where you want to save the test report, and then click Finish.

4 Select the server where the services have been deployed.

5 Enter valid login credentials.

6 Select the Service Context where the services have been deployed.

7 You can choose to view the report by pressing the View Report button. Or, access the report by opening the specified output directory using Windows Explorer.

**Note:** By default, all M3 API calls will return a maximum of 100 records. To override this setting, use the "Max Records" setting for the specific test case in the Test Editor.

## Deleting a Test Case

1 Select the test case that you want to delete from the table and click Delete.

A confirmation message is displayed prompting you to confirm the deletion.

2 Click Yes.

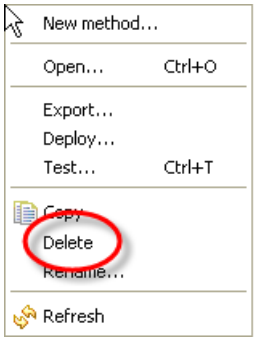
The test case is removed from the list.

## Deleting a Web Service

Use this procedure to delete a web service that is no longer needed. Deleting a web service from the M3 Web Services Designer removes only the metadata of the web service from the repository. It does not remove the web service from the M3 Web Services Server. The web service will still be available for consumption; however, you will not be able to work with it in the M3 Web Services Designer.

To remove a web service from the M3 Web Services Server, you must undeploy the web service.

1 Select the web service that you want to delete and right-click it.




2 Select Delete to delete the web service.

A message is displayed prompting you to confirm the deletion.

3 Click Yes.

The web service is deleted from the Web Service Repositories tab.

## Undeploying a Web Service

Use this procedure to undeploy a web service.

1 In the Servers view, right-click on the web service and choose Undeploy.

2 Choose Yes to confirm.

The web service is undeployed from the server.

# Working with Web Service Methods

5

This section describes how you work with web service methods.



## Web Service Methods Overview



### What are Web Services Methods?

Web services expose their functionality through methods. Consumers call these methods when they want to request the service that the method provides through the web service.

M3 Web Services Designer allows you to create methods for different types of programs.

If you are using M3 Business Engine, you can use the following:

- M3 API Programs. For more information, see "Creating a Method Wrapping an M3 API program" on page 41.
- M3 Display Programs. For more information, see "Creating a Method Wrapping an M3 Display Program" on page 36.

You can also write web services for SQL Queries. For more information, see "Working with SQL Queries" on page 44.

You can add multiple methods to a web service.

Methods are displayed in the Web Service Repositories view. To view methods associated with a web service, click to expand the web service. Methods are displayed underneath web services in the tree view. The type of icon next to the method provides error and warning information:

- A green icon indicates that the method has no errors or warnings.
- A yellow triangle indicates that then method has warnings, but can be deployed.
- A red square indicates that the method has errors and is not deployable.

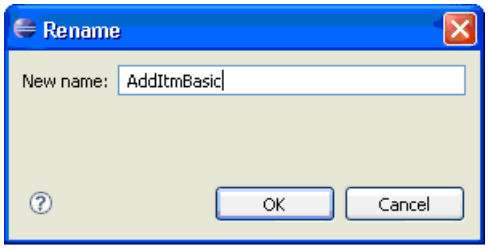
# Working with Methods

### To rename a method

1 Select the method that you want to rename and right-click.

2 Select Rename to rename the method.

The Rename window is displayed.




3 Type the new name for the method in the New name field and click OK.

The new name appears in the Web Service Repositories view.

### To delete a method

1 In the Web Service Repositories view, right-click on the method that you want to delete.

2 Select Delete.

A message is displayed prompting you to confirm the deletion.

3 Click Yes.

The method is deleted from the Web Service Repositories tab.

**Note:** When you delete a method from a web service, it is removed only from the M3 Web Services Designer, that is, the metadata related to the method is removed. To remove the method from the M3 Web Services Server, you need to save the web service after deleting the method, and re-deploy the web service.

## Copying a Method

Use this procedure to copy a method from one web service to another.

1. In the Web Service Repositories view, expand the repository and the web service where the method that you want to copy is located.
2. Select the method that you want to copy and right-click.
3. Select Copy to copy the method.

   The selected method is copied to the clipboard.
4. In the Web Service Repositories view, select the repository and the web service where you want to copy the method to and right-click.
5. Select Paste.

   The method is now copied to the new repository location.

# Working with M3 Display Programs



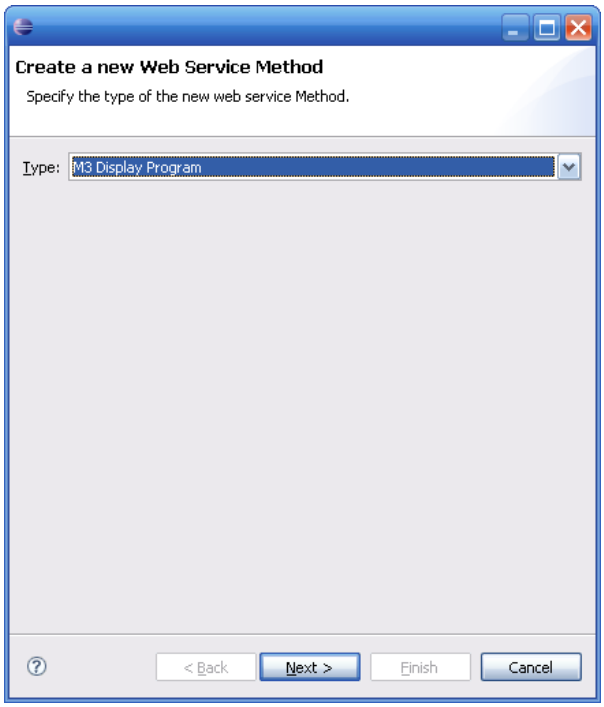
## Creating a Method Wrapping an M3 Display Program

### To create a method wrapping M3 Display Program

1. In the Web Service Repositories view, expand the repository location where the web service is located for which you want to create a method.
2. Select the web service for which you want to create a method and right-click.

**3** Select New method.

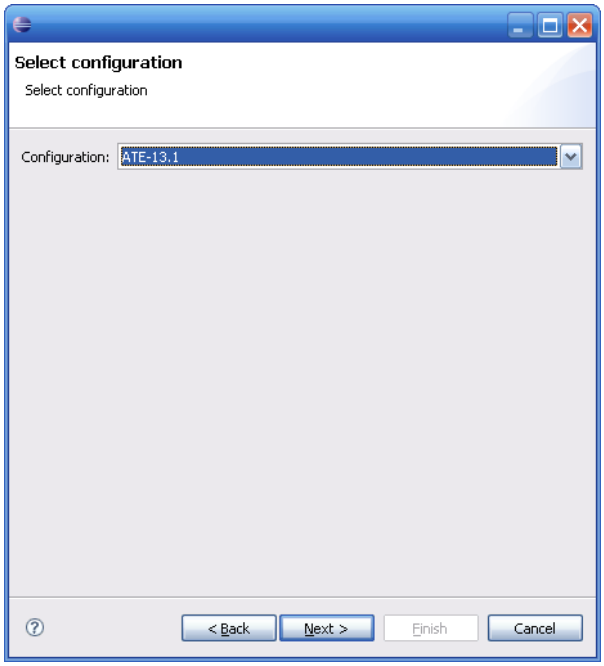
The Create a new Web Service Method window is displayed.




**4** Select M3 Display Program from the drop-down list and click Next.

**5** Select a server from the drop-down list and click Next.

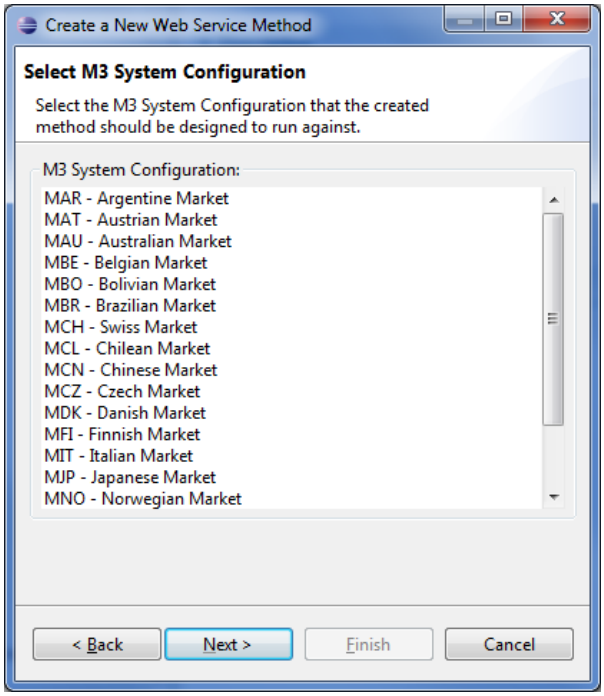
**6** Select the configuration that is applicable for the M3 Display Programs, then click Next.




**7** Enter M3 Server View user name and password, then click Next.

**Note:** Your M3 Server View user name and password are similar to your regular M3 login credentials.

A list of system configurations is shown.

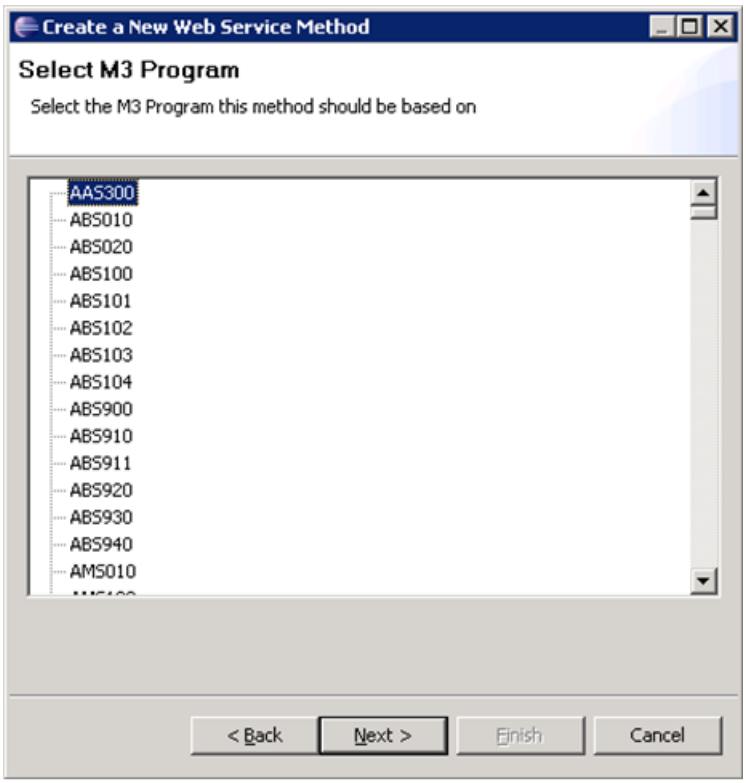



**8** Select a system configuration.

System configuration determines the order of fetching metadata for the web service method creation by looking in the view definition sub directories and in the standard view definition directory. In addition, all applied fix pack directories are searched for in the correct order.

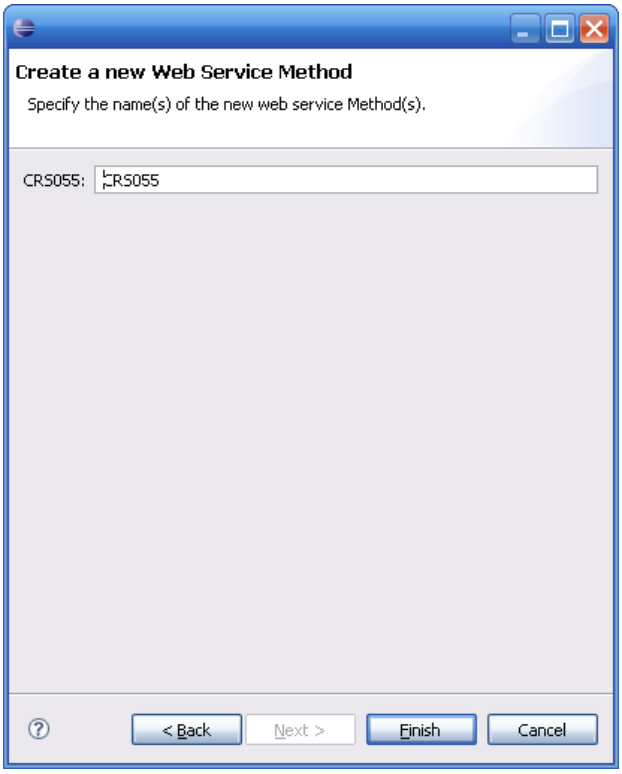
**9** Click Next.

The Select M3 Program window appears.




**10** Select from the list of M3 programs.

**11** Enter a meaningful name that describes the functionality. For example, use **`CRS055`** , then click Finish.




The method name must conform to the XML naming rules.

**Note:** Fore more information, go to **`http://www.w3.org/TR/REC-xml/`**.

The method is displayed in the Web Service Repositories view and the details in the M3 Display Program editor on the right side. For more information on using the M3 Display Program, see "Using M3 Display Program Editor" on page 66

**Note:** A warning may be displayed indicating that certain programs may be considered unsuitable to be used for web services. Examine the warning message (click the warning text for a detailed description) and determine what is most suitable in your case. Click Continue or Cancel.

## Adding a Related Program

A related program is a program that is being called from the main program. A related program is one that receives its input for further processing from the main program (or another related program).

For example, if you want to automate the process of printing a confirmed customer order, you need to first create a method that wraps the M3 Display Program OIS100. OIS100 is used to create customer orders. Next, associate the program OIS605 to automatically print the confirmed customer order. OIS605 is an example of a related program which when chained with the main program OIS100 gives rise to a single web service method.

1 Open the method to which you want to add a related program.

The details are displayed in the editor area.

2 Select the main program (in this example OIS100) and click Add Related.

The Select Configuration window is displayed.

3 Select a configuration, and then click Next.

4 Enter M3 Server View user name and password, and then click Next.

**Note:** Your M3 Server View user name and password are similar to your regular M3 login credentials.

A list of system configurations is shown.

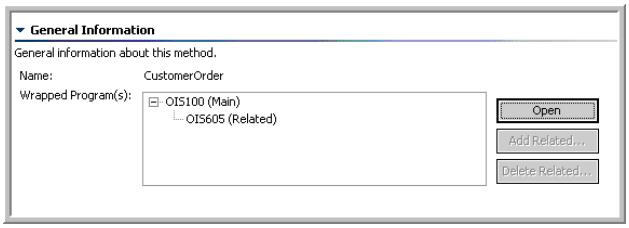
5 Select a system configuration.

6 Click Next.

A list of M3 programs appears.

7 Select the called program (in this case, OIS605) and click Finish.

The program is displayed in the Wrapped Programs list.




8 Open the related program and specify the panel sequence, input fields and output fields.

# Working with M3 API programs


- "Creating a Method Wrapping an M3 API program" on page 41


## Creating a Method Wrapping an M3 API program

M3 API consists of two distinct parts: the Client Interfaces and the M3 Interfaces.

- The Client Interfaces are the low level functions that are used in different environments to access the M3 Interfaces.
- The M3 Interfaces consist of the transaction descriptions and the MI (M3 Interface) programs in the Application Server.

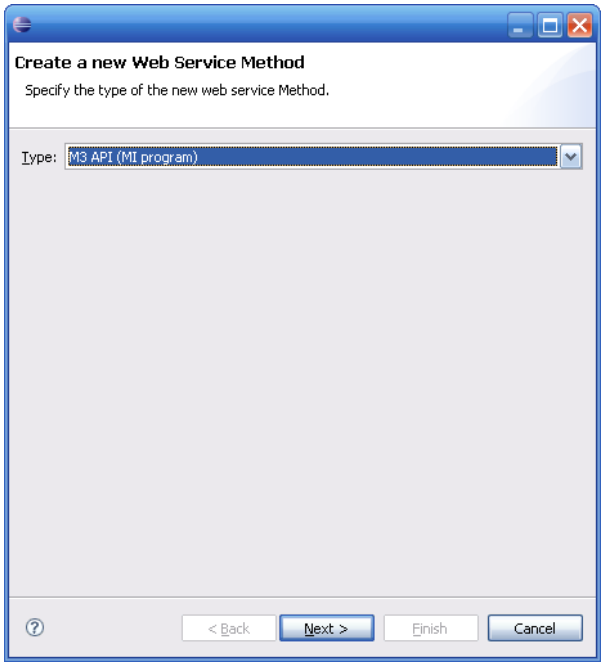
## To create a method wrapping an M3 API program

1 In the Web Service Repositories view, expand the repository location where the web service is located for which you want to create a method.

2 Select the web service for which you want to create a method and right-click.

3 Select New method.

The Create a new Web Service Method window is displayed.

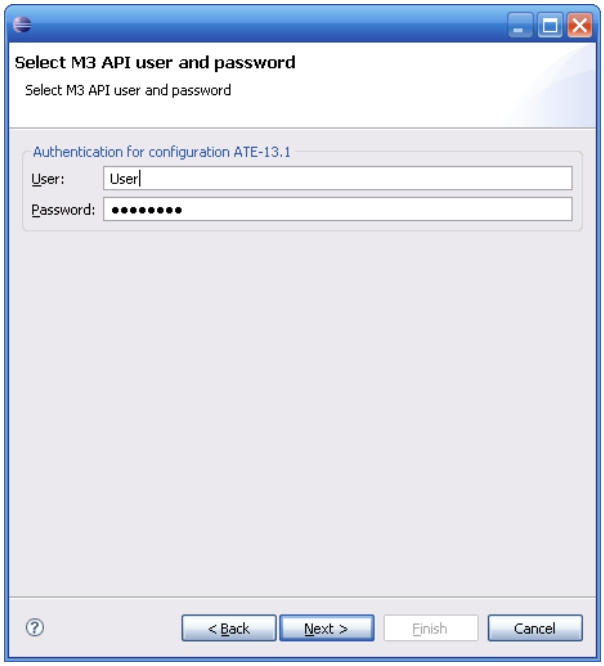



4 Choose ‘M3 API (MI program)’ from the Type drop-down list and click Next.

5 Select a server from the drop-down list and click Next.

6 Select the applicable configuration from the list of configurations and click Next.

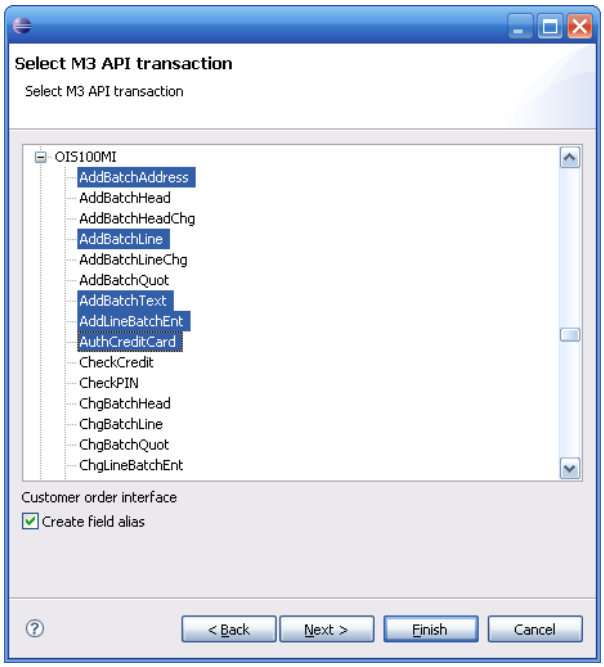
The Select M3 API user and password window is displayed.




**7** Provide your M3 server user name and password to authenticate the configuration that you selected in the previous step and click Next.

A list of M3 API programs is displayed.

Select a program and expand it to choose the transaction for which you want to create the method.

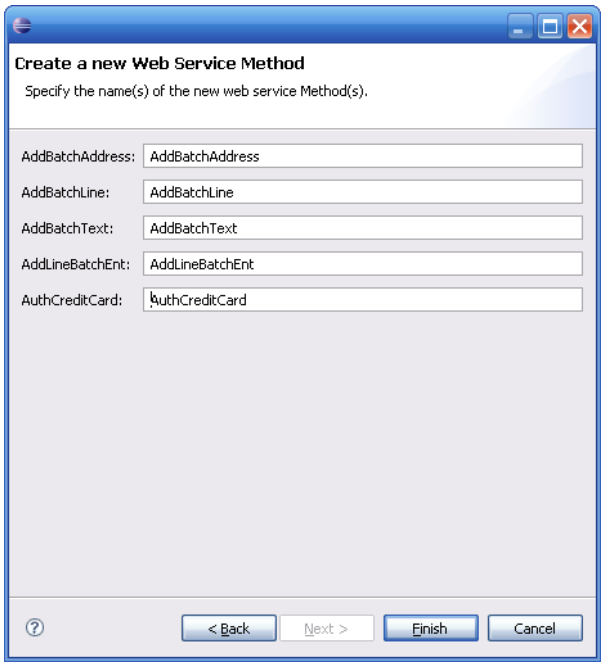


**Note:** As shown in the picture above, it is possible to select multiple transactions, and a method for each transaction will be created.

8 Choose the checkbox Create field alias, if you want the aliases for the field name to be displayed in the Alias column of the API Editor.

For more information about Aliases, see "Common functions" on page 75.

9 Provide a name for the method in the Name field and click Finish.




It is recommended that you provide a meaningful name that describes the functionality. For example, AddBatchLine, AddBatchText, AddLineBatchEnt, AuthCreditCard.

The name must conform to the XML naming rules. For more information, go to **`http://www.w3.org/TR/REC-xml/`**.

Click Finish.

The method is displayed in the Web Service Repositories view and the details are displayed in the editor on the right side. For more information on the M3 API Editor, see "M3 API Program Editor" on page 73

# Working with SQL Queries

This section describes how you work with web services using SQL Queries.

## Creating an SQL Configuration

Before creating web services based on SQL, you must add an SQL configuration to the M3 Web Services Server.

**Before you start** If you are planning to create web services that make SQL calls, you must install a JDBC driver corresponding to the database that the web service will access. For more information, see the *M3 Web Services Administration Guide*.

## Creating a Method Wrapping a Single SQL Query

Sometimes it can be desirable to present data directly from a database via a web service. This can be useful if you have an application that does not provide any APIs or support for web services. To handle this scenario, M3 Web Services provides a web service wrapper over JDBC based SQL statements.

**Note:** This is an advanced feature, and knowledge of both SQL and the database model of the database being accessed are required.

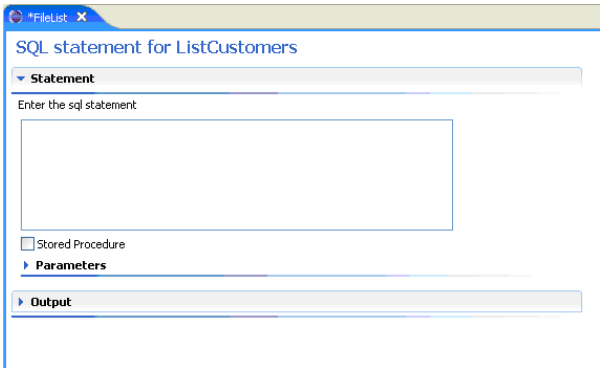
In JDBC, SQL statement input parameters are mapped to the question marks in the statement. The number and order of the input parameters should always match the number of question marks.

```
SELECT OrderNo FROM Orders WHERE Date > ?
```

**Note:** This requires that the input parameter representing the date is added to the list of input parameters.

1. In the Web Service Repositories view, expand the repository location where the web service is located for which you want to create a method.
2. Select the web service for which you want to create a method, right-click and select New Method. The Create a new Web Service Method window is displayed.
3. Select type SQL Query from the drop-down list and click Finish.

   The SQL Editor is opened.




The editor is split into two sections, each containing a single subsection:

- Statement

  This is where the SQL statements are entered.

  Statement > Parameters - If the statement has any ‘variable’ input parameter, this is where they are described.

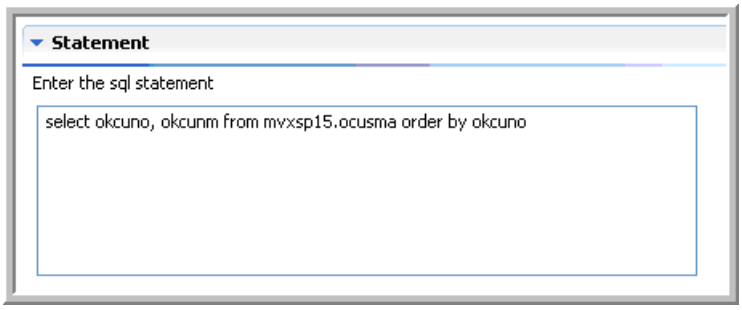
- Output

  This is where the output from the queries is described.

  Output > Result Set - Result sets are defined for the output of each of the queries defined in the statement section.

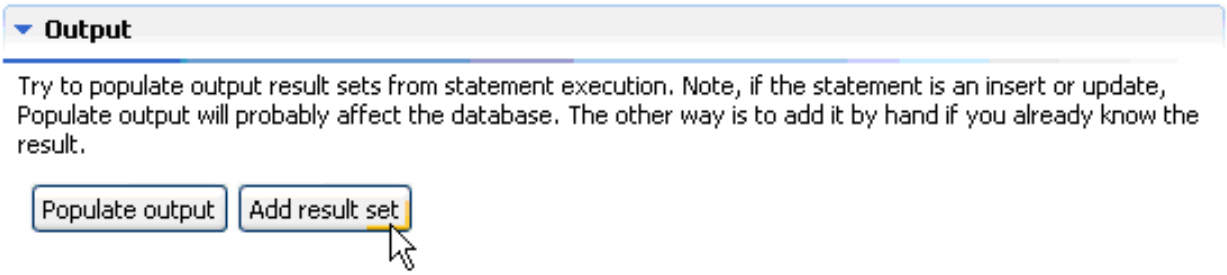
**4** Make a simple list of customers in an M3 database, using an SQL ‘select’ statement.

**Tip:** At this point it is necessary to have knowledge of the tables and columns that will be used in your query.




**5** Click Output to create a result set for the output.

The Output section is expanded.




6 Create a result set. There are two options:

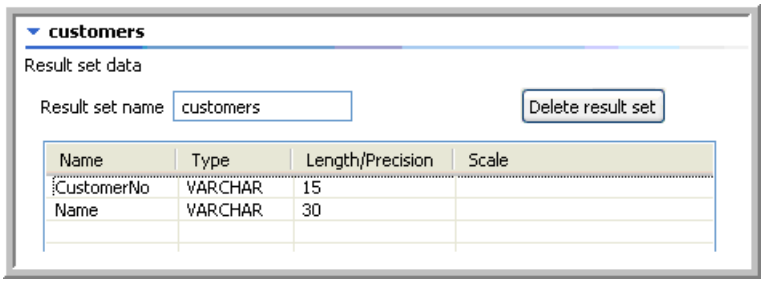
- Click Populate output to automatically create a result set. For this to work, a local (client-side) installation of the JDBC driver being used is required. If you click Populate output, continue with section "Populating Output" on page 48.
- Click Add result set. If you click Add result set continue with section "Adding an Output Result Set" on page 47.

## Adding an Output Result Set

This procedure describes how to add results to the Output section.

1 Click Add result set.

The Result set data section is expanded.

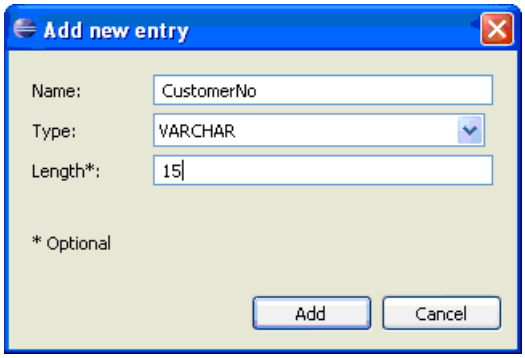



2 Enter a name for the Result set.

In this example, we have used the name **`customers`**.

3 Add a result set entry, click New.

The Add new entry is displayed.



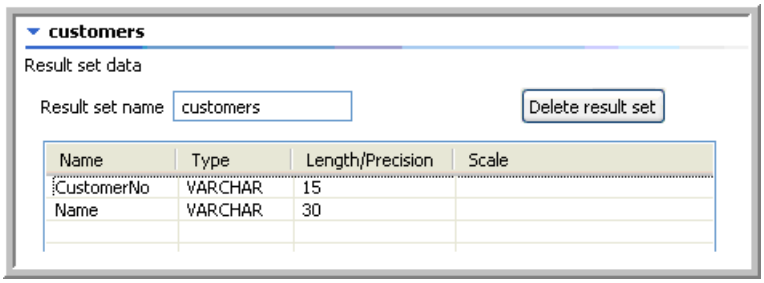

**4** Enter the name and type of the entry to contain the output, and then click Add.

**Note:** The length is not required. In most cases it should be left out but it depends on the application consuming the service to a certain extent.

In this example, there are two outputs, okcuno and okcunm, and therefore entries for each of these is needed. It is important to get these in the same order.

The name of the field can be anything you like (spaces are not allowed), but the type should match the field type from the SQL statement.

The field types are JDBC data types, and may not match what you are used to in your specific database. In the main, the numeric fields are easy to identify, but string or character fields are usually represented as type VARCHAR (variable length character field).




# Populating Output

In order to automatically create a result set, the M3 Web Services Designer must be able to call the database itself. To do this, it requires a JDBC driver. The JDBC driver is usually a jar file available from the database vendor's website. If SQL queries are enabled on the server side, you can ask your M3 Web Services Server administrator for the correct driver to use, as the server also requires the same file.

For more information about importing the required JDBC driver, see "Specifying Development Settings" on page 17.

### To populate output

1 Click Populate output from statement.... The Select configuration window is displayed.

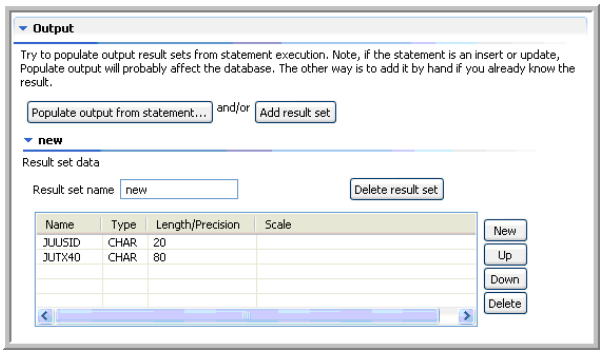
2 Select the configuration so that the correct driver name and URL can be determined. Click OK.

The Enter database credentials window is displayed.

3 Enter the login to the database.

Click Finish.

The output results are now retrieved.




**Note:** The above example from M3 shows the field lengths as double the length you might expect. This is a quirk of the driver and a Unicode database, and is not a fault of M3 Web Services.

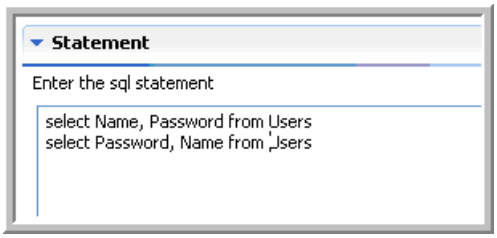
4 Save the result by pressing CTRL+S.

## Creating a Method Wrapping Multiple SQL Queries

M3 Web Services supports multiple statements within the same method. This could be useful if, for example, you want to insert a row into a database table with a SQL insert, and then check that it got inserted correctly by following the insert with an SQL select.

1 Perform steps 1 – 3 in section "Creating a Method Wrapping a Single SQL Query" on page 45. The SQL Editor is opened.

2 Enter the SQL statements. The Statement input box support multiple statements separated by a line break like in the example.

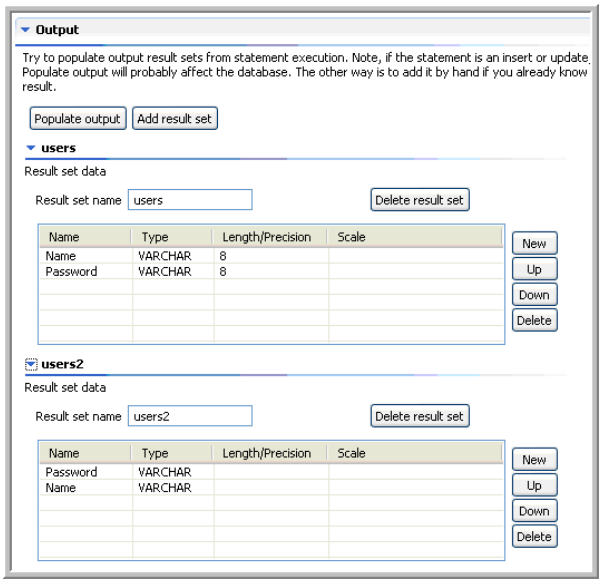



**3** These queries do not require any variable input parameters so we only need to create a result set for the output. Click Output.

The Output section is expanded.

**4** Click Add result set.

The Result set data section is expanded.




**Note:** The Length/Precision is optional, there is no reason why it must be specified on one result set and not the other in the example shown.

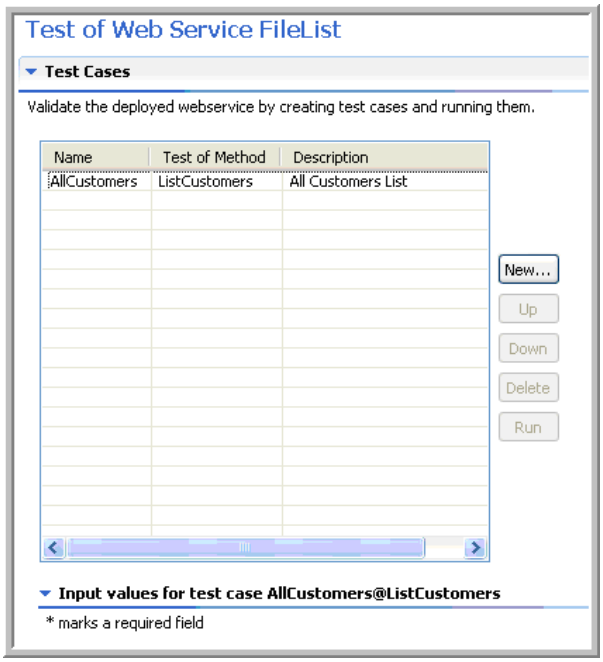
# Creating a Test Case

**Need More Details?** Check out the following concepts:

- "Testing a Web Service" on page 25.

**1** Click the Test Tab link in the Overview tab and click New. The Create a new test case window is displayed.

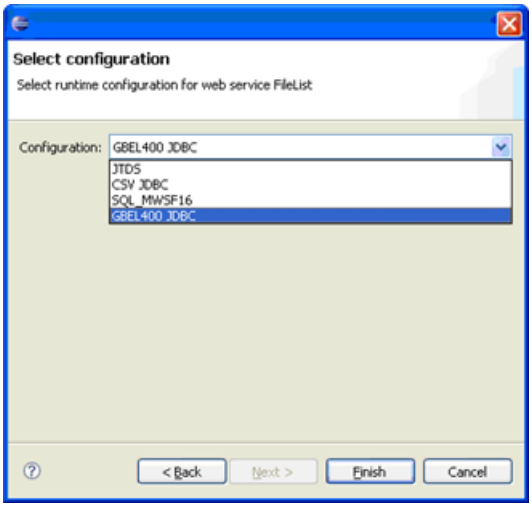
2 Enter the method and add a name and description. Click Finish. The Test Cases editor appears.




3 As we need no input to this test case, we are done.

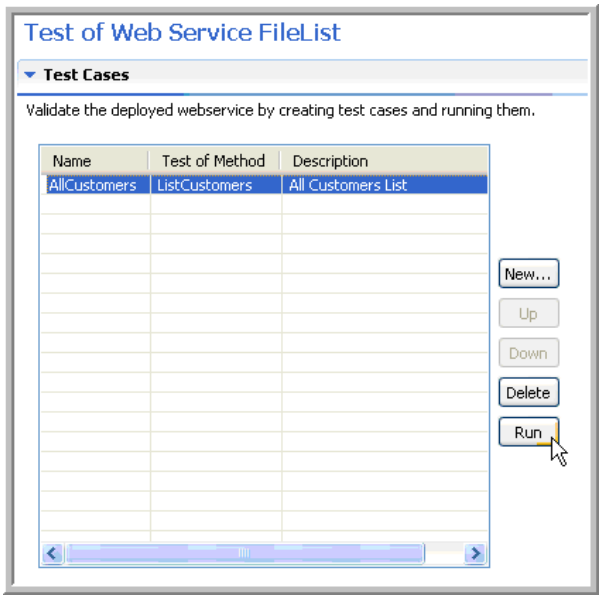
4 Select Deploy from the context menu of the web service.

5 Select the applicable server followed by the applicable configuration (example).



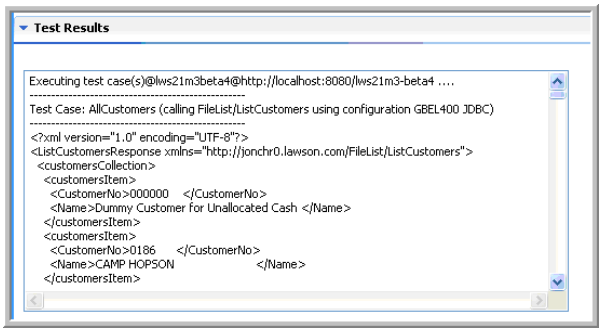

6 Click Finish.

7 Return to the test case.




**8** Click Run to run the test case.

**9** After you have entered the correct user/password for the database as requested, the results should be shown. Note how the names for the elements match those that we keyed on our result set earlier.




# Creating and Calling SQL Stored Procedures

M3 Web Services also supports the calling of Stored Procedures. Store Procedures is considered to be an advanced topic, and differs from database to database, therefore, it is difficult to document. If you want to use Stored Procedures, it will be necessary to consult the documentation for your database and JDBC driver.

However, the basic SQL syntax for calling a stored procedure via JDBC is as follows: **`{call MY_PROCEDURE}`**

Below, you find a partial example of creating and calling a stored procedure on i5/iSeries which lists an M3 customer table as in our other examples. First, the stored procedure is created.

**Tip:** There are many ways to create the stored procedure, and there are IBM Redbooks and manuals that cover this in more detail.

1 On the i5/iSeries, we used the SQL command to create the Stored Procedure:

```
Create procedure cgjlib/CUST_LIST result sets 1 language SQL BEGIN declare c1 cursor with return for select okcuno, okcunm from mvxsp15/ocusma; open c1; END
```

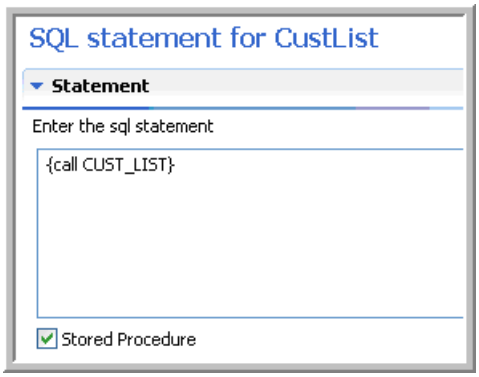
The Stored Procedure is named CUST_LIST, and is placed in the library CGJLIB. This library must be defined within the libraries parameter of the JDBC driver URL in the server configuration.

2 In the Web Service Repositories view, expand the repository location where the web service is located for which you want to create a method.

3 Select the web service for which you want to create a method, right- click and select New Method.

The Create a new Web Service Method window is displayed.

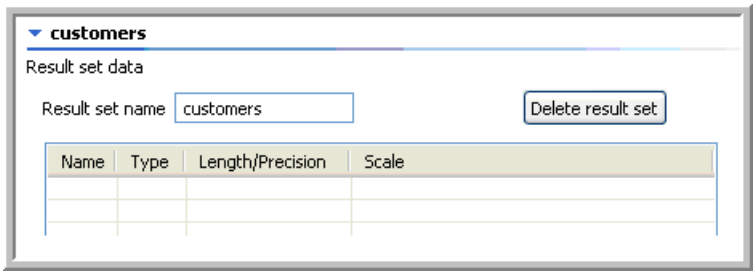
4 Select type SQL Query from the drop-down list and click Finish. The SQL Editor is opened.



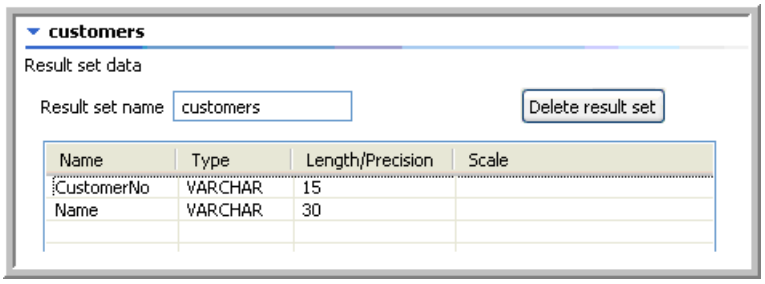

5 Enter the SQL Statement and make sure that the Stored Procedure checkbox is selected.

6 Click Output. The Output section is expanded.

7 Click Add result set. The Result set data section is expanded.




8 Enter a name for the Result set. In this example, we have used the name new.




**9** Add a result set entry, click New.

**10** Enter the name and type of the entry to contain the output and click Add.

**Note:** There are two outputs, okcuno and okcunm. Entries for each of these are needed. It is important to get these in the same order.

The name of the field can be anything you like (spaces are not allowed), and the type should match the field type from the SQL statement. The field types are JDBC data types, and may not match what you are used to in your specific database. In the main, the numeric fields are easy to identify, but string or character fields are usually represented as type VARCHAR (variable length character field).

**11** Deploy the web service.

**12** Create and run a test case as described in section "Testing a Web Service" on page 25.

# Web Services and Security

6



## Running a Secure Web Service

M3 Web Services supports the use of WS-Policy, which is based on public key cryptography, also known as asymmetric cryptography. The implementation is based on signing and encrypting SOAP messages handled by the SOAP engine CXF.

### Authentication: Key Exchange Process

M3 Web Services can be run in secured mode. This is enabled in the M3 Web Services Server View under Manage Service Security. If secured mode is enabled, the following applies to the communication between clients and the M3 Web Services Server:

**Important:** You cannot use the Web Services Designer Test feature to test secured web services. You must use a third party application.

For more information in implementing secured web services, see "Using X.509 Policy to Secure Web Services" on page 77.

- The M3 Web Services Server generates a private/public key pair at installation that is unique to this M3 Web Services Server. In order for clients to be able to call the secured web services they need some way to identify them. This means that a client needs a public/private key pair and also that the M3 Web Services Server has imported the client's public key as a trusted certificate.
- When the client has created its keys, the certificate (public key) needs to be uploaded to the Server. This ensures that the M3 Web Services Server only trusts valid client certificates. The client needs to get the certificate of the M3 Web Services Server and incorporate it into its keystore. This certificate is needed for encrypting messages.
- When using the delivered x509 security policy, the message sent from the client will be signed and encrypted. This ensures that the M3 Web Services Server knows who the message is from and that the message content is safe from outside view.

**Note:** For the M3 Web Services Server to be able to authenticate an incoming message it needs to be signed with a valid certificate. This client-certificate must be known to the M3 Web Services Server. Only the clients that have uploaded their certificate will be authenticated.

## Client <-> M3 Web Services Server

The X509 Policy is supplied with M3 Web Services to sign and encrypt the SOAP message using X.509 certificates.

## M3 Web Services Server <-> back-end

Back-end systems are M3 or databases. The user credentials are supplied by HTTP layer basic authentication.

# Migrating Web Services

7

This section describes how to migrate web services from older versions to the current.



## Migrating Older Versions of Web Services

The following migration paths are supported.

### MWS 1.6 Web Services to M3 Web Services Designer

You can convert web services created in Movex Web Services 1.6 to web services that can be used in the current version of M3 Web Services Designer.

- Only the conversion of web services that wrap the M3 Display Programs are supported. The web services that wrap the Application Program Interface (API) can be easily re-created in M3 Web Services Designer.
- A converter runs in the background to facilitate the migration process. The converter fetches the metadata that is not available for the MWS 1.6 web service from the M3 Server that you connected to while creating the MWS 1.6 repository, adds it to the web service, and then converts it to an M3 Web Services web service.

  A difference between MWS 1.6 and M3 Web Services is that aliases for field names are automatically used when you have a correct configuration.

### Migrating Lawson Web Services 2.1 Web Services

No specific procedures are needed. Add your old repository to your new M3 Web Services Designer and the web services are ready to be used or edited.

**Note:** Web services that have been edited in later versions can no longer be used in Lawson Web Services 2.1.x.

### Migrating Lawson Web Services 9.0.x Web Services

No specific procedures are needed. All web services created in earlier 9.0 versions are fully compatible with this version. Add your old repository to your new M3 Web Services Designer and the web services are ready to be used or edited.

## Migrating MWS 1.6 Web Services to M3 Web Services Designer

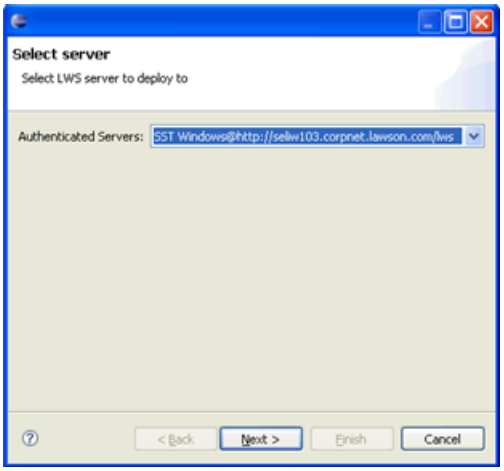
**Before you start** To complete this procedure, you must have access rights to the MWS 1.6 Server.

If you have not been assigned a user name and password for the MWS 1.6 Server, please contact your MWS Server administrator.

### To migrate an MWS 1.6 service

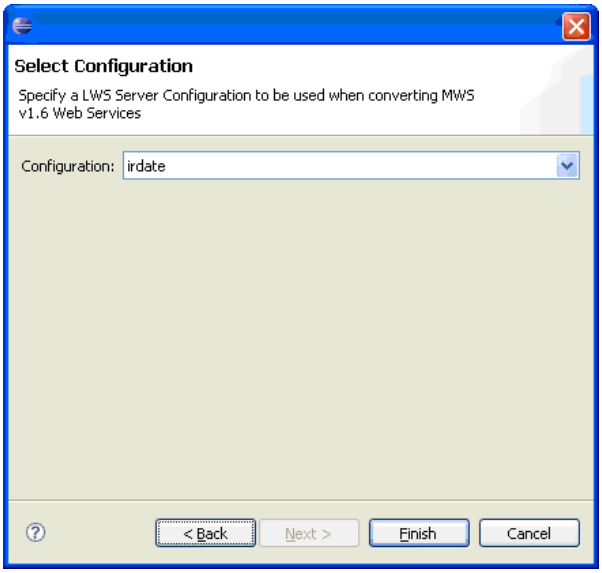
1. In the Servers tab, click to expand an MWS 1.6 server.
2. Select a web service. Drag it to a web services repository.

   The Select Server window is displayed.




3. Select a server with an M3 configuration and click Next.

   The Select Configuration window is displayed.



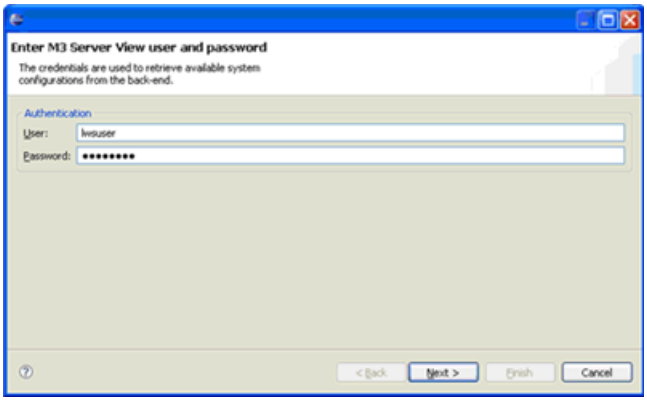

**4** Select an M3 configuration.

The configuration details are dependant on the M3 Web Services Server that you have chosen in the previous step.

By choosing the M3 Server, in this step you are creating a connection to the specified M3 Server in order to fetch the metadata.

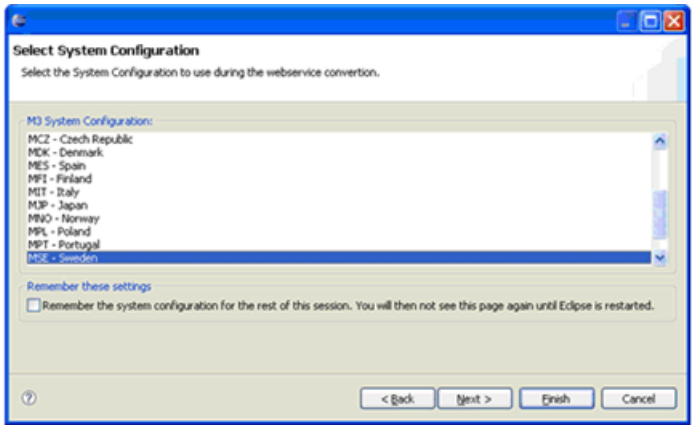
The metadata that was not available in the MWS 1.6 Web Services will be retrieved from this M3 Server while migrating the web services to run in M3 Web Services Server.

**5** Enter back-end server view user name and password.



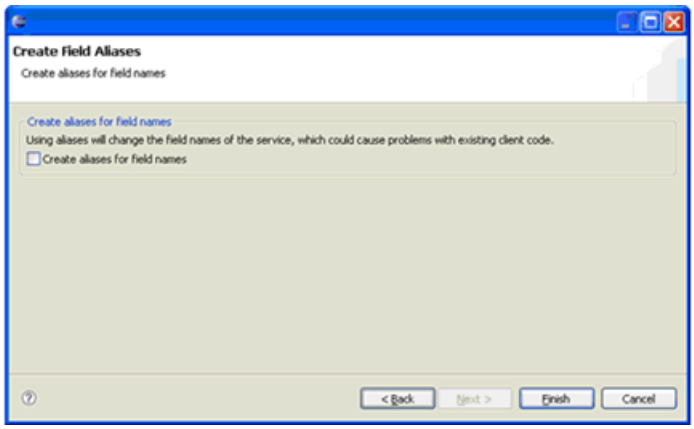

Click Next.

**6** Select the system configuration that you want to use in fetching additional metadata required by M3 Web Services.




**7** Click Next.

The Create Field Alias window is displayed.




**8** Check the Create aliases for field names checkbox if you want to automatically fetch aliases for the fields in the method.

**Important:** You could change your web services clients to use aliases instead in order for the web services to work. If you, however, still wish to use field names in your old or new web services clients, you could uncheck the Create aliases for field names checkbox.

Click Finish.

# Server Management and Troubleshooting

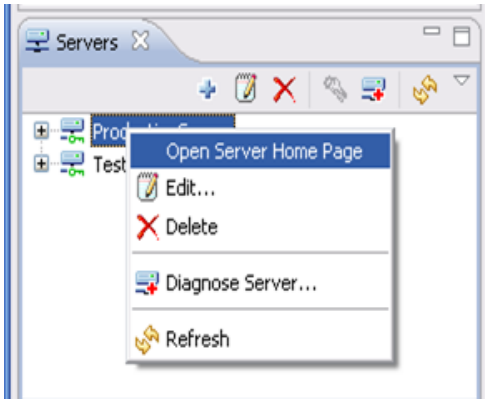
8



## Accessing the M3 Web Services Server Management Pages

The M3 Web Services Server management pages in the Grid are the administrator interface for the M3 Web Services servers.

- In the Server view, right-click on the server and choose Open Server Home Page




The M3 Web Services Server Home Page appears. From here, you can launch the Management pages, which will be opened using Java Web Start.

For more information on this interface, see the *M3 Web Services Administration Guide*.

# Diagnosing M3 Web Services Designer

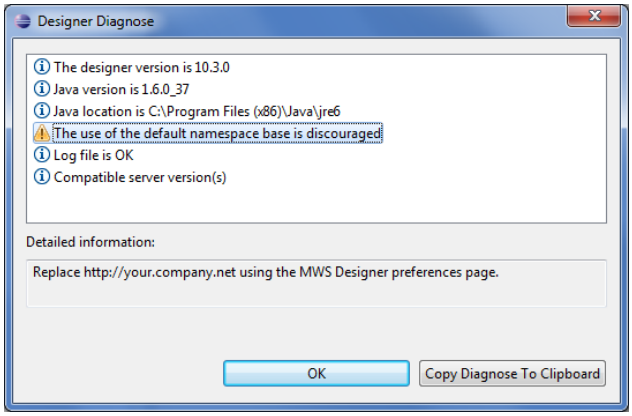
Use this procedure to perform a diagnostic check on M3 Web Services Designer.

**1** From the Eclipse menu bar, choose MWS Designer> Designer Diagnose.

The Designer Diagnose window is displayed.

To display more information, click a warning or an error message.

To send this information along to a support organization, click the warning or error, and then click Copy Diagnose To Clipboard and insert the information in an e-mail or error report.




**2** Click OK to apply the settings and exit the Server Diagnostics window.

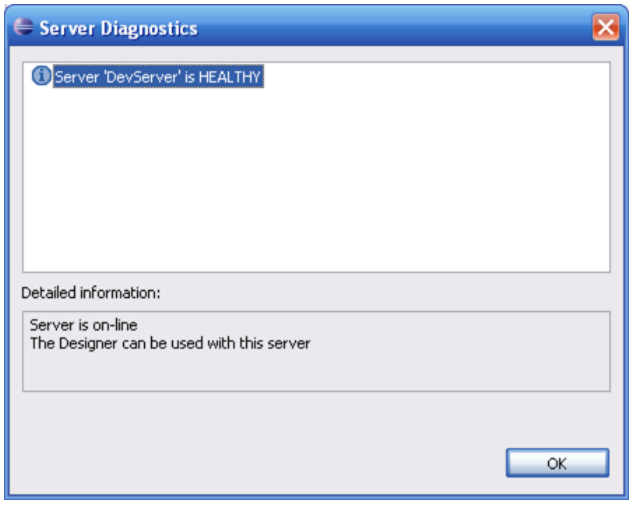
# Diagnosing the M3 Web Services Server

Use this procedure to diagnose an M3 Web Services Server. The diagnostic information includes warnings, error messages, and information messages.

**1** Go to the Server tab.

**2** Right-click on the server you want to check and choose Diagnose Server.

The Server Diagnostics window is displayed.




To display more information, click on a warning or on an error message.

**3** Click OK to close the window.

## Viewing Designer Log Files

The View Designer Log can be used to display the contents of the log file, which is useful for support personnel to troubleshoot the M3 Web Services Designer.

**Tip:** When M3 Web Services Designer is put in debug mode the amount of log entries can be too exhaustive to be useful. Use the Clear log and close … button to clear the log. For information on how to change the logging levels, see "Specify Log File Settings" on page 18

## Specifying Communication Settings

Use this procedure to specify settings for the communication between the M3 Web Services Server and the Designer.

The Connection timeout (s) and Execution timeout (s) are by default set to 60 seconds.

**Important:** Do not change this value unless you have timeout problems.

### To change communication settings

**1** From Eclipse, choose Window>Preferences.

**2** Expand MWS Designer and choose Communication Settings.

3 Complete the following fields:
   - Connection timeout (s)
   - Execution timeout (s)

4 Click Apply.

5 Click OK to apply the settings and exit the Preferences window.

### To reset cached MWS server login information

1 Click the Clear Cached Passwords button.

2 Click OK.

## Error Message: Java out of Memory

If a web service contains many methods, the deployment may fail due to the error "Java out of memory". In order to recover from this error situation, you need to change the settings for the Java Runtime Environment.

You need to edit the shortcut to Eclipse that you created upon installation of the M3 Web Services Designer:

```
ECLIPSE32_INSTALL_DIR>\eclipse.exe -vmargs -Xmx768M
```

The argument "-vmargs" above allocates 768 MB of memory for the runtime environment. The amount of memory to allocate differs depending on your needs and the size of your available memory. You may need to experiment to find a suitable maximum heap size to allocate to match the needs for your largest web service.

You can also set the heap size in the Eclipse Preferences window.

### To set the heap size

1 Navigate to Window > Preferences.

2 On the left panel, select Java > Installed JREs.

3 On the right panel, select the compiler, and then click Edit.

   The Edit JRE window opens.

4 Specify the heap size in the Default VM Arguments input field.

   For example, if you enter **`-Xms250m -Xmx700m" value for the "Default VM arguments`** in the Default VM Arguments input field, the compiler will have a minimum heap size of 250 MB and a maximum heap size of of 700 MB.

**Tip:** You can split the web service into two and divide the methods.

**5** Click OK.

# Using M3 Display Program Editor


This section describes the use of the M3 Display Program Editor.



## M3 Display Program Editor

The M3 Display Program editor provides the User Interface (UI) through which you can define the Panel sequence and the output fields, and provide input for the web services based on M3 Display Programs.

The M3 Display Program is run using the inputs entered through the editor. The resulting output is obtained in XML format. This allows seamless exchange of data between applications.

The editor also provides the following functionality:

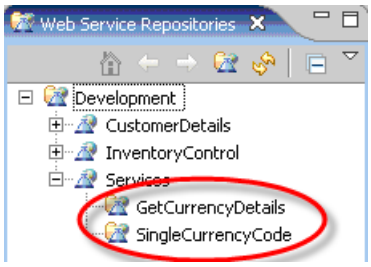
- Add related programs
- Delete related programs

You can open the editor in the following two ways:

- In the Web Service Repositories view, select the web service that contains the method wrapping the M3 Display Program and double-click. The overview of the web service is displayed in the editor area.


Select the method in the Methods section and click Open. The editor is displayed.

- In the Web Service Repositories view, expand the web service that contains the method. Select the method and double-click. The editor is displayed.

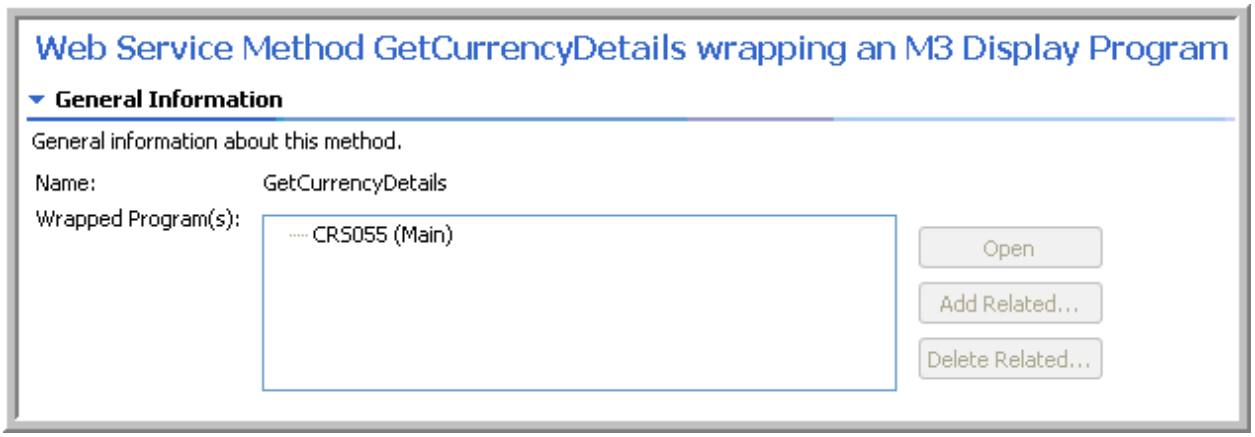



# Using M3 Display Program Editor

The M3 Display editor can be broadly divided into two sections:

- General Information

  Contains general information about the method wrapping the M3 Display Program. It also contains three buttons – Open, Add Related, and Delete Related.




- Open

  Expands the section that contains fields through which you can gather input data for the web service, define output fields and the panel sequence.

- Add Related

  Allows you to add related programs that are associated with the main program.

- Delete Related

  Deletes the related program.

- M3 Program

  This contains the fields and options through which you can provide data for the method.


  The options under the M3 Display Program are categorized into the following three sections:

  - Panel Sequence

    This is where you specify the panel sequence for the method. The sequence specified here determines the available input and output parameters. The panel selected should be typically the one that is used when the specific program is run in the M3 Net Extension (MNE).

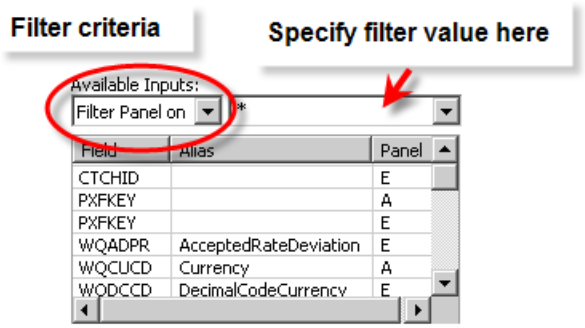
  - Input to *program*

    This is where you specify the input fields for the method. This contains two tables:

    Available Inputs

This table contains all the available fields, both required and optional, that a selected M3 Display Program has on all its panels.

You can choose to filter the fields list on three filter criteria – Fields, Alias and Panels.

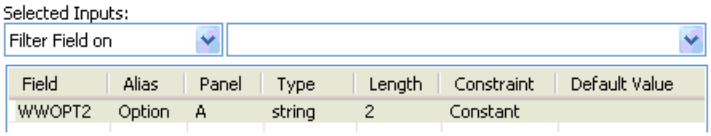



Choose the filter criterion from the drop-down list and enter the value. For example, if you want to display only those fields that contain the character 'W', then choose the criterion, Filter Field on, and type **W** in the text box.

Selected inputs

This contains all the fields that user will use in the web service to get the data for a successful execution of the web service.

**Note:** Make sure that you have selected all the mandatory and constant input fields from the Available Inputs for a successful execution of the web service.




You can also choose to filter the fields list based on any column heading.

- Output from *program*

  This is where you specify the output fields. The output fields are the results that a web service publishes after successful execution. The result should be the same as the output of the selected M3 Program when it is run in Infor Smart Office.

  **Note:** This section also contains two tables and offers the filtering features similar to that of the section "Specifying the Input Fields" on page 71.

## Specifying a Panel Sequence

**1** Select one of the following alternatives:

- Select the program in the Wrapped Program(s) list and click Open.
- Expand the M3 Program section

**2** Click New to specify the panel sequence for this method.

The Add a new panel window is displayed.

**3** Choose A from the Panel drop-down list.

The first panel should always be A, which is the ‘key’ panel in M3. If the M3 program does not have an A panel, the B panel can be used.

**Note:** Either A or B panel should to be selected as the first and last panel in the majority of circumstances. The panel sequence should not contain both A and B panels.

**4** Select one of the following alternatives:

- Choose Enter to move to the next panel from the Function Key drop-down list.
- Choose a key that captures the task that you want to execute.

A function key is a special key on the keyboard (F1, F2, etc.) that is assigned a specific task and is used as a short cut to execute that task.

The Function Key is used here in order to automate tasks such as clicking Next to move to the next panel.

The list of function keys differs based on the panel that you have chosen.

These function keys are available in the action menu of the M3 Display Program. (Run the program in Infor Smart Office and place the mouse on the Actions menu to see the list of Function Keys available for each panel.)

**5** Click Finish.

The panel is added in the Panel sequence table.


**6** Follow steps 2 – 6 to add the other panels. Make sure that you end the Panel sequence with the F3 (Exit) at the end of the panel sequence.

At the end of this task all the panels are listed in the Panel sequence table.

**Note:** If you want to change the sequence of the panel at a later stage, you can do so by moving the panels up and down using the Up and Down buttons.

**7** To delete a panel, select the panel and click Delete.

# Specifying the Input Fields

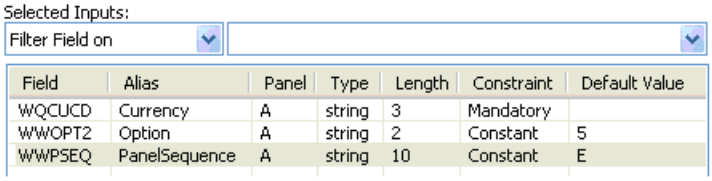
**1** Select the field that you want as an input field from the Available Inputs table.

You can select multiple fields by holding the Ctrl key and the left mouse button.

**2** Click the >> button to add an input field in the Selected Inputs table.

All the selected fields are displayed in the table.

Click the << button to remove a field from this Selected Inputs table.




| Field | Alias | Panel | Type | Length | Constraint | Default Value |
|---|---|---|---|---|---|---|
| WQCUCD | Currency | A | string | 3 | Mandatory | |
| WWOPT2 | Option | A | string | 2 | Constant | 5 |
| WWPSEQ | PanelSequence | A | string | 10 | Constant | E |

**3** Select the Constraint type for from the list.

Constraint is one of the following:

- Mandatory

  For fields that are part of primary keys or required input
- Optional

  For fields that can be input or left out
- Constant

  For fields that should always be a specific known value and not input by the user

**4** Double-click in the Default Value column to provide a default value for the field.

The fields ‘WWOPT2’ (Option) and ‘WWPSEQ’ (Panel Sequence) should be provided with a default value.

The common values for option are:

- 1 = Create
- 2 = Change
- 3 = Copy
- 4 = Delete
- 5 = Display.

The field WWPSEQ (Panel Sequence) should match that of the flow you want to reproduce from Infor Smart Office. Refer to Infor Smart Office for the correct values.

**Note:** Panels A-D should not be used in the Panel Sequence input even if they occur in the flow.

# Specifying the Output Fields

1. Select the field that you want as an output field from the Available Outputs table.

   You can select multiple fields by holding the Ctrl key and the left mouse button.
2. Click the >> button to add an output field in the Selected Outputs table.

   All the selected fields are displayed in the table.

   Click the << button to remove a field from the Selected Outputs table.
3. Use CTRL+S or File > Save to save the changes.

# M3 API Program Editor

B

This section explains the use of the M3 API Program Editor.

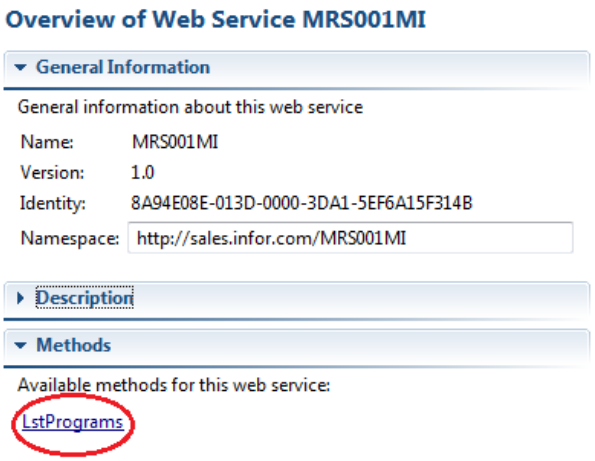


## Introduction

The M3 API Editor provides the User Interface to create web services for API programs in M3 Web Services Designer. It also allows you to test the web services through the Test tab. See "Testing a Web Service" on page 25.

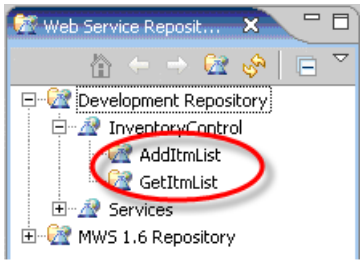
You can open the editor in the following two ways:

- In the Web Service Repositories view, select the web service that contains the method, which wraps the M3 API program and double-click. In this example, the web service InventoryControl is chosen. The overview of the web service is displayed in the editor area.




  Select the method in the Methods section and click Open. The editor is displayed in the right-hand side of the M3 Web Services Designer perspective.

- In the Web Service Repositories view, expand the web service that contains the method. Select the method and double-click. The editor is displayed in the right-hand side of the M3 Web Services Designer perspective.

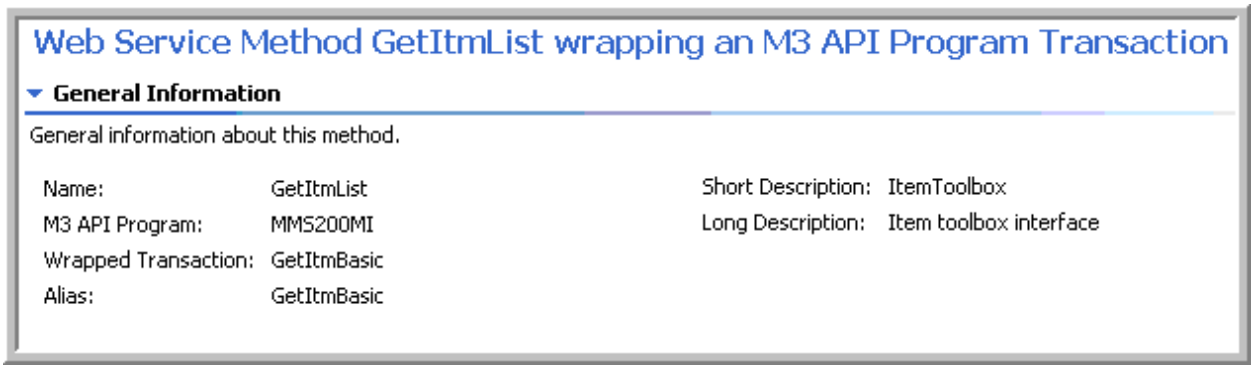



## Using the M3 API Editor

The M3 API editor can be divided into two sections:

- General Information

  This contains general information about the method that wraps the M3 API program.




  The fields in the General Information are as follows:

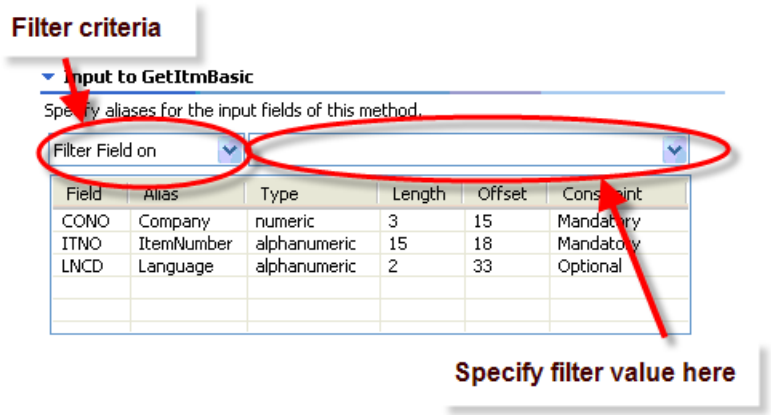
| | |
|---|---|
| **Name** | Name of the web service method |
| **M3 API Program** | Name of the M3 MI/API program that is wrapped by the web service |
| **Wrapped Transaction** | Transaction in the M3 MI/API program that is wrapped by the web service method |
| **Alias** | Alias of the transaction in the MI/API program |
| **Short Description** | Short description of the program |
| **Long Description** | Long description of the program |

- The second section contains two tables – Input to *program* and Output from *program*.


Input to *program*

This table contains all the available fields, both required and optional fields that a selected M3 MI/API program has on all its panels. You can edit the aliases for the input fields.

- Mandatory are required input that M3 programs need in order to work.
- Optional inputs are not required.
- You can edit the Type for the input parameters




**Note:** The aliases are displayed only if you have selected the Create field alias checkbox while creating the method. For detailed information, see "Creating a Method Wrapping an M3 API program" on page 41.

- Output from *program*

  The output fields are the results that a web service publishes after successful execution. The result should be the same as the output of the selected M3 MI/API program when it is run in the Infor Smart Office.

  You must select which fields you want returned to the calling client. All output fields are optional, but without any output fields, a warning is given. For more information about specifying output fields, see "Specifying the Output Fields" on page 72.

## Common functions

The following are the common functions to both the Input and Output tables:

- Filtering the contents

You can filter the contents of the tables based on any heading (column name) in the tables. The drop-down box (indicated by the right-hand side arrow in the figure) contains the filter criteria. Choose the filter criterion from the drop-down list and enter a value. For example, if you want to display only those fields that contain the letter ‘W’, then chose the criterion Filter Field on, remove ‘*’, and type ‘W’ in the text box.

- Sorting the values

  You can sort the values in the ascending or descending order by clicking on the heading (column name). For example, if you want to sort the values in the column Type, click on the column heading Type.

- Editing alias column

  You can edit the Alias column of both the tables. This provides you with the flexibility of changing the default name to something that is suitable to the customer’s requirements or to even translate to other languages. For example, the alias for ITNO could be changed to ‘ProductNumber’ from ‘ItemNumber’.

  **Note:** Make sure that there are no spaces in the Alias name.

- Opening multiple methods

  You can open multiple methods for editing in the editor. Each method is represented as a tab at the bottom of the editor. However, only one tab is active at any given time.

# Using X.509 Policy to Secure Web Services

C



## Using X.509 Policy to Secure Web Services

X.509 policy is an alternative to https (SSL) where the main purpose is to secure the SOAP message. It supports timestamp, signature and encryption. It implements the concept of PKI (Public Key Infrastructure).

These instructions assume you are using Java with CXF. For more information, see http://cxf.apache.org/

### Creating a keystore and key

Use the Keytool.exe cmd in the Java SDK environment to generate a keystore and a key.

**Note:** You can type "Keytool -help" to list all available parameters for Keytool. More info about Keytool can be found here http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/keytool.html

1 Generate a keypair. Make sure the "-keyalg RSA -sigalg MD5withRSA" is present.

```
Keytool -genkeypair -keystore keystoreClient.jks -storepass secretPWD
-alias myalias -keyalg RSA -sigalg MD5withRSA -dname "c=SE, cn=John Doe, o=Infor,
st=Sweden, l=Linköping"
```

2 Export the certificate:

```
Keytool -exportcert -keystore keystoreClient.jks -storepass secretPWD
-alias myalias -file myalias.cer
```

3 Import the certificate into MWS. Open the MWS administration home page and click Upload Client Certificates. Browse to locate the exported certificate file. To upload it to the server, click Add.

4 Import the MWS Server certificate into your keystore. On the MWS administration home page, click Download Server Certificate. Choose where to store the certificate file. Import it to your keystore by running the following Keytool command.

```
Keytool -importcert -keystore keystoreClient.jks -storepass secretPWD
-alias MWSserver -file MWSServerCert.cer
```

## Using keys and certificates in client code

In your client code, you make use of these keys and certificates for signing and encrypting with X.509 tokens. Before making the call to the actual web service, you must add "interceptors" to handle the X.509 policy.

### Example: using interceptors

```
private void add_interceptors() throws SOAPException, IOException {

    Bus b = BusFactory.getDefaultBus();

    // Note, remember to remove the interceptor from the bus if you want to make
    // another call from the same (running) jvm without using X509 policy

    addX509TokenOutInterceptor(b.getOutInterceptors());

   // handles outgoing message (i.e. request)

    addX509TokenInInterceptor(b.getInInterceptors());

   // handles incoming message (i.e. response)
```

### Example: addX509Token... methods

```
// Add a WSS4JOutInterceptor with X509 Token to given interceptor list
  private void addX509TokenOutInterceptor(List<Interceptor> list){

    // alias for key the will be used for signing the message, key with that
    // alias (private) needs to exist in the client keystore
    String user = "myalias";

    // alias cert will be used for encryption, must have been imported to the
    // client keystore
    String encryption_user = "MWSserver";

    Map<String, Object> securityProperties = new HashMap<String, Object>();

    // Make sure the secClient.properties file and keystore is on the classpath

    // OUT (Adds a timestamp and signs and encrypts the outgoing message)
    securityProperties.put(WSHandlerConstants.ACTION, WSHandlerConstants.TIMESTAMP
       + " " + WSHandlerConstants.SIGNATURE + " " + WSHandlerConstants.ENCRYPT);
    securityProperties.put(WSHandlerConstants.SIG_PROP_FILE, "secClient.properties");
    securityProperties.put(WSHandlerConstants.ENC_PROP_FILE, "secClient.properties");
    securityProperties.put(WSHandlerConstants.PW_CALLBACK_CLASS,
      ClientCallbackHandler.class.getName());

    securityProperties.put(WSHandlerConstants.USER, user);

    securityProperties.put(WSHandlerConstants.ENCRYPTION_USER, encryption_user);
    list.add(new WSS4JOutInterceptor(securityProperties));
  }

  // Add a WSS4JInInterceptor with X509 Token to given interceptor list
  private void addX509TokenInInterceptor(List<Interceptor> list){
    // IN
    Map<String, Object> inProps = new HashMap<String, Object>();

    inProps.put(WSHandlerConstants.ACTION, WSHandlerConstants.TIMESTAMP
        + " " + WSHandlerConstants.SIGNATURE + " " + WSHandlerConstants.ENCRYPT);
    inProps.put(WSHandlerConstants.SIG_PROP_FILE, "secClient.properties");
    inProps.put(WSHandlerConstants.DEC_PROP_FILE, "secClient.properties");
    inProps.put(WSHandlerConstants.PW_CALLBACK_CLASS,
        ClientCallbackHandler.class.getName());
    inProps.put(WSHandlerConstants.ENABLE_SIGNATURE_CONFIRMATION, "false");

    WSS4JInInterceptor wssIn = new WSS4JInInterceptor(inProps);
    list.add(wssIn);
```

```
}
```

The ClientCallBackHandler class serves the password for the alias used when signing (the password for the keypair generated by Keytool).

**Example: ClientCallbackHandler implementation**

```
public class ClientCallbackHandler implements CallbackHandler {

  public void handle(Callback[] callbacks) throws IOException,
    UnsupportedCallbackException {

     WSPasswordCallback pc = (WSPasswordCallback) callbacks[0];

     // set the password for our message.
     if (pc.getIdentifier().equals("myalias")){
        pc.setPassword("myaliaspwd");
   // return the password for the key for the signing alias in the client keystore
     }
     else if (pc.getIdentifier().equals("nisse")){
        pc.setPassword("efj7an4aa");
     }
  }
```

The secClient.properties file contains information about the keystore.

**Example: secClient.properties file**

```
org.apache.ws.security.crypto.merlin.keystore.password=secretPWD
org.apache.ws.security.crypto.merlin.keystore.type=JKS
org.apache.ws.security.crypto.merlin.file=keystoreClient.jks
org.apache.ws.security.crypto.provider=org.apache.ws.security.components.crypto.Merlin
```

After making these modifications, your code should contain all necessary security tokens.

## Troubleshooting

If you get an error message, try the following:

- Check the MWS Server log (located in the Grid administration page for MWS).
- If you get an "Security processing failed"message with no further explanation, change the log level for MWS/CXF to INFO (instead of default ERROR). Try the operation again and examine the log files for detailed information.