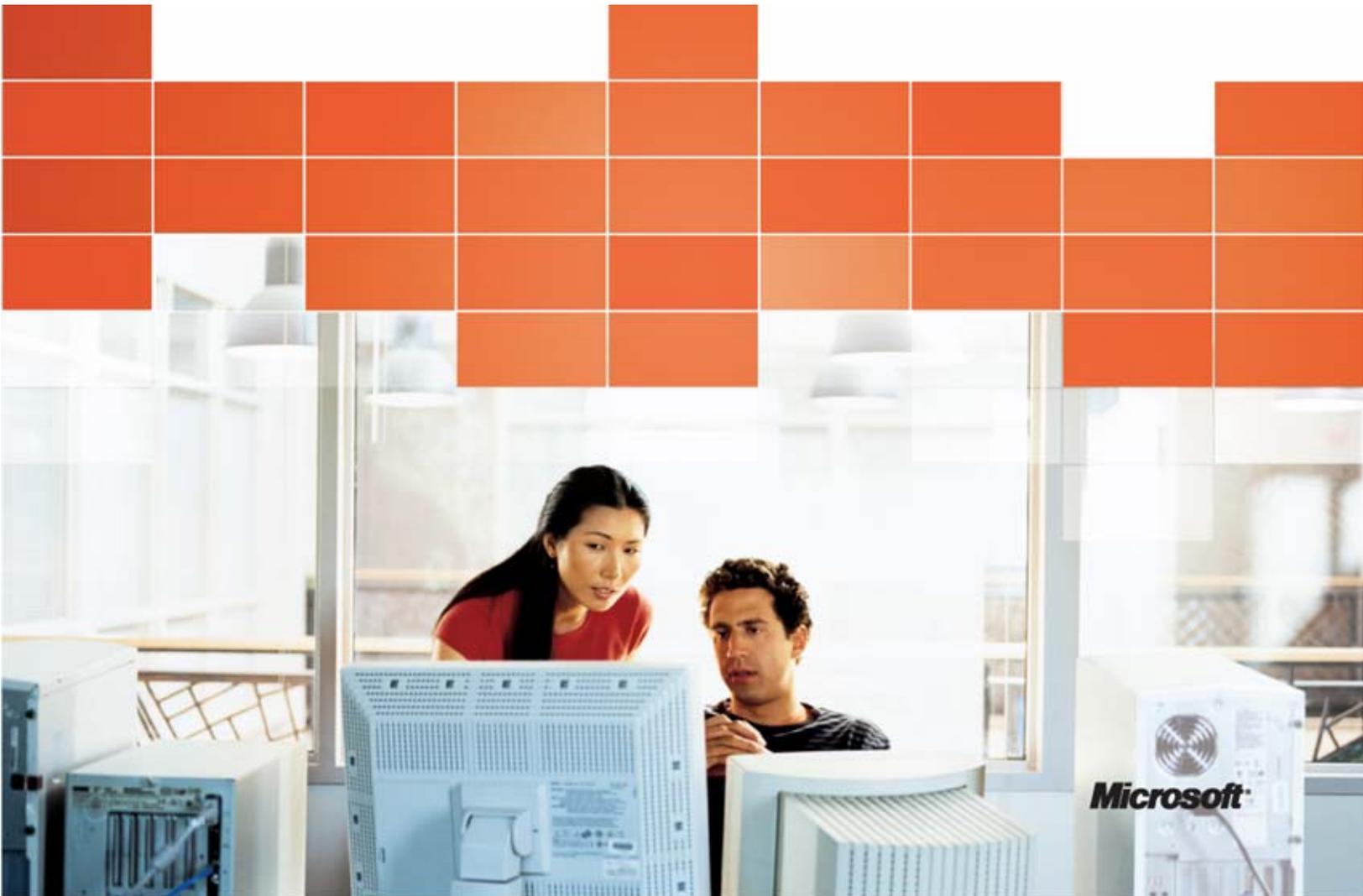




## Creating a MOM 2005 Event Report

Last Reviewed:	October 17, 2005
Product Version:	MOM 2005
Reviewed By:	Justin Harter
Latest Content:	<a href="http://spaces.msn.com/members/jharter">http://spaces.msn.com/members/jharter</a>



# Introduction

This guide demonstrates how to create a simple report for MOM 2005. While creating a simple report does not require advanced knowledge of Visual Studio, it will be EXTREMELY useful to have some knowledge of Visual Studio.

## Prerequisites

To create a MOM 2005 report using this guide, you must have the following items installed:

### SERVER

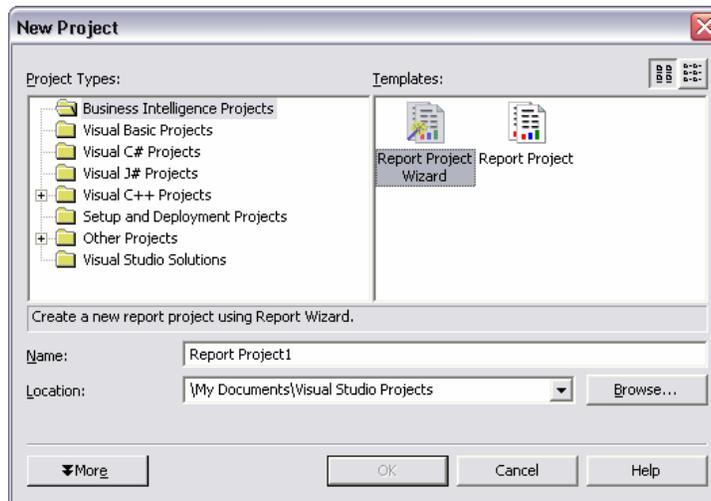
- SQL Reporting Services (server)
- MOM Reporting (server)

### CLIENT

- SQL Reporting Services (client components)
- Visual Studio .NET 2003

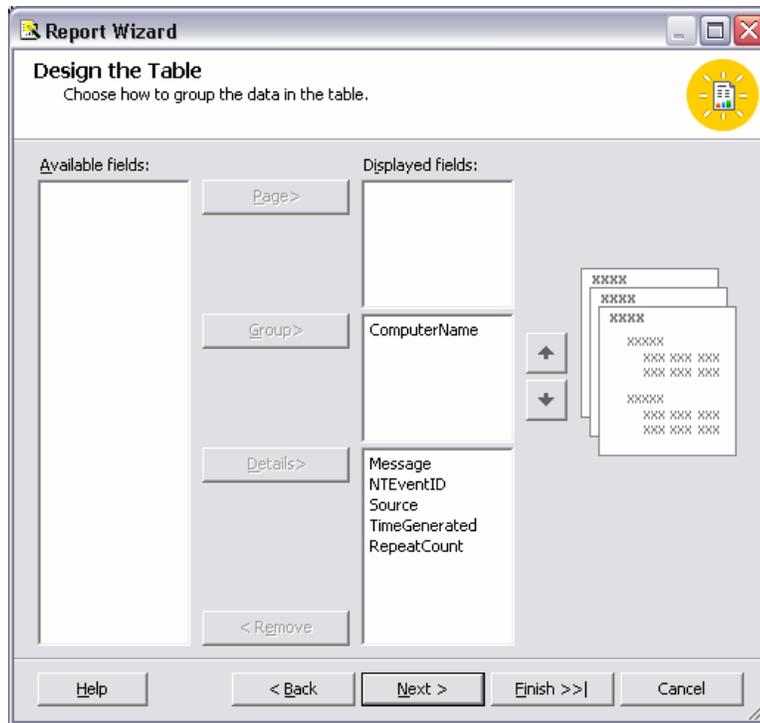
# Creating a Report Using the Wizard

1. Begin by opening Visual Studio .NET 200
2. Click File → New → Project
3. If you have installed SQL Reporting Services client components, you will see a project type of “Business Intelligence Projects”. Double click on the “Report Project Wizard” template.

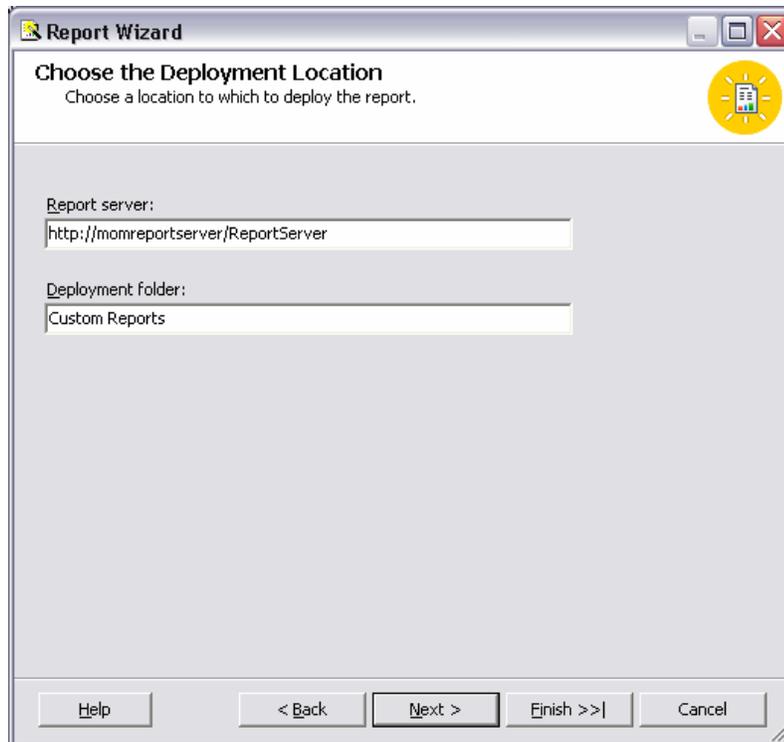


4. Once the report wizard appears, click Next.
5. Next we need to configure the data source. To do this, click on the “Edit” button. This will bring up a dialog in which you must fill out the three areas (Server Name, Credentials, & Database). You need to point the data source to the MOM reporting server and the SystemCenterReporting database. For the credentials, I generally use the same account that I used for my other MOM reports. To check this go to the MOM reporting console and edit the existing SCDW data source. It will show you the credentials used. The DAS account would be a good choice for this. You may not be able to specify these credentials in the MOM wizard (I believe it will only accept SQL credentials). As a result, you may have to add the credentials after you post them to the MOM reporting site. In the meantime, simply use an integrated Windows account that has access.
6. Next, you will be prompted for the query string. The MOM SDK provides views for many queries. You can read more about these views in the SDK documentation. A common report which is asked about is for a particular event. In this instance, I will use the following query:
 

```
Select ComputerName, Message, NTEventID, Source, TimeGenerated, RepeatCount
FROM SDKEventView
```
7. For report type, simply choose “Tabular”.
8. Next, you will need to design the table layout. You can group the events within the report by computer by following the screenshot below.



9. Next, choose "Stepped" for the table layout.
10. Next, choose whatever table style you'd like to use. In this example, I used Compact.
11. Next, choose the deployment location.



12. Finally, type a name for the report and click Finish.

After the report has been created, you will need to customize it so that it is dynamic and user friendly. You will need to do various visual modifications depending on the report you create and you will also need to create some parameters so that your report is dynamic (Begin Date, End Date, Server, Event ID, etc.).

13. Next, I will increase the text box size for the Message field. By default, the wizard created a field that was very small and cumbersome to read in a report.

	Computer	Message	NTEventID	Source	Time	Repeat Count
	=Fields!Comput					
		=Fields!Messag	=Fields!NTEvent	=Fields!Source.	=Fields!TimeGe	=Fields!RepeatC

14. Next, I will add a parameter for the Event ID and the Date so that this report can be dynamic. The next step is optional, but it will make your report more user friendly. We will add a dataset so that we can populate the default value for the date (such as the last week). Click on the "Data" tab and click on the dataset dropdown box and choose "New Dataset". Fill in the information accordingly.

The screenshot shows the 'Dataset' dialog box with the following fields and values:

- Name:** \_currentDate
- Data source:** SCDW
- Command type:** Text
- Query string:**

```
SELECT *
FROM fn_GetDateRange(7) fn_GetDateRange_1
```
- Timeout:** (empty)

15. Click on the "Report" menu within Visual Studio and choose "Report Parameters". Click the "Add" button.
16. Create a parameter similar to the information listed in the screenshot below for the BeginDate, EndDate, and EventID:

**Report Parameters**

Parameters:

- BeginDate
- EndDate

Properties:

Name: BeginDate Prompt: Begin Date

Data type: DateTime

Allow null value

Allow blank value

Available values:

Non-queried

From query

Label	Value
*	

Default values:

Non-queried

From query

None

Dataset: CurrentDate

Value field: BeginDate

Buttons: Add, Remove, OK, Cancel, Help

**Report Parameters**

Parameters:

- BeginDate
- EndDate

Properties:

Name: EndDate Prompt: End Date

Data type: DateTime

Allow null value

Allow blank value

Available values:

Non-queried

From query

Label	Value
*	

Default values:

Non-queried

From query

None

Dataset: CurrentDate

Value field: EndDate

Buttons: Add, Remove, OK, Cancel, Help

**Report Parameters**

Parameters:

- BeginDate
- EndDate
- EventID

Properties:

Name: EventID Prompt: Event ID

Data type: String

Allow null value

Allow blank value

Available values:

Non-queried

From query

Label	Value
*	

Default values:

Non-queried

From query

None

Buttons: Add, Remove, OK, Cancel, Help

17. Now that the parameters have been created, you must tell the report query to use the parameter. Click on the Data tab and change the query for the SCDW Dataset to be:

```
Select ComputerName, Message, NTEventID, Source, TimeGenerated, RepeatCount
FROM SDKEventView
WHERE (TimeGenerated > @BeginDate) AND (TimeGenerated < @EndDate) AND
NTEventID = @EventID
```

18. Now we are ready to preview our report. Click on the "Preview" tab and fill in the necessary information (Begin Date, End Date, Event ID). As you can see, the report will by default use the last week as the Begin Date and End Date. To change this modify the CurrentDate dataset by putting in a number other than 7.

The screenshot shows the SSRS report preview window. The 'Preview' tab is selected. The 'Begin Date' is set to 10/10/2005 1:11:39 PM and the 'End Date' is 10/17/2005 1:11:39 PM. The 'Event ID' is 21241. The report title is 'Event Report'. The table below shows the data:

Computer Name	Message	NTEventID	Source	Time Generated	Repeat Count
TESTMOM1	The MOM Server detected a change to the rules for one or more computers, and will begin downloading the new rules and configuration settings to the affected computers.	21241	Microsoft Operations Manager	10/10/2005 2:40:48 PM	
TESTMOM1	The MOM Server detected a change to the rules for one or more computers, and will begin downloading the new rules and configuration settings to the affected computers.	21241	Microsoft Operations Manager	10/10/2005 2:45:59 PM	
TESTMOM1	The MOM Server detected a change to the rules for one or more computers, and will begin downloading the new rules and configuration settings to the affected computers.	21241	Microsoft Operations Manager	10/10/2005 2:51:13 PM	

19. When you are satisfied with the report, we are ready to deploy it to the MOM reporting site. First we must specify where we want the report to be published. To do this, open up the report properties by going to Project menu → <report name> Properties. Change the target folder to something like Custom Reports. This is a recommended practice so that you do not accidentally lose your custom reports due to a change in the directory structure or an MP change.

The screenshot shows the 'Event Report Property Pages' dialog box. The 'Configuration' is set to 'Active(Debug)' and the 'Platform' is 'N/A'. The 'Deployment' section is expanded, showing the following properties:

Property	Value
StartItem	Event Report.rdl
OverwriteDataSources	False
TargetFolder	Custom Reports
TargetServerURL	http://TESTMOM1/ReportServer

The 'TargetServerURL' property is highlighted. Below the table, there is a description: 'TargetServerURL: The URL of the report server to which to deploy the project, for example, http://hostname/ReportServer.'

- Finally, click the Build menu and choose "Deploy Solution". You should now be able to navigate to the MOM reporting site and find your custom report.

Another useful parameter, which is commonly used, would be the Computer Group parameter and the Computer parameter. This way you could do a search for a specific computer with the event ID you specify. To do this, simply create 2 more datasets with the information shown below. Then create 2 parameters with the information shown below.

The screenshot shows the 'Dataset' configuration window for a dataset named 'CompGroup'. The 'Data source' is 'SCDW', the 'Command type' is 'Text', and the 'Query string' is:

```
SELECT *
FROM fn_ListComputerGroups() fn_ListComputerGroups_1
ORDER BY CompGroup
```

The 'Timeout' field is empty. The window has 'OK', 'Cancel', and 'Help' buttons at the bottom.

The screenshot shows the 'Dataset' configuration window for a dataset named 'ComputerList'. The 'Data source' is 'SCDW', the 'Command type' is 'Text', and the 'Query string' is:

```
SELECT Computer
FROM fn_GetComputersInGroup(@CompGroup)
fn_GetComputersInGroup
order by Computer
```

The 'Timeout' field is empty. The window has 'OK', 'Cancel', and 'Help' buttons at the bottom.

The screenshot shows the 'Report Parameters' configuration window for the 'CompGroup' parameter. The 'Name' is 'CompGroup' and the 'Prompt' is 'Computer Group'. The 'Data type' is 'String'. The 'Available values' are set to 'From query', with the 'Dataset' set to 'CompGroup', 'Value field' set to 'CompGroup', and 'Label field' set to 'CompGroup'. The 'Default values' are set to 'None'. The window has 'Add', 'Remove', 'OK', 'Cancel', and 'Help' buttons.

The screenshot shows the 'Report Parameters' configuration window for the 'Computer' parameter. The 'Name' is 'Computer' and the 'Prompt' is 'Server'. The 'Data type' is 'String'. The 'Available values' are set to 'From query', with the 'Dataset' set to 'ComputerList', 'Value field' set to 'Computer', and 'Label field' set to 'Computer'. The 'Default values' are set to 'None'. The window has 'Add', 'Remove', 'OK', 'Cancel', and 'Help' buttons.

## Using an Existing Report as a Template

Sometimes it is easier to start from an existing report and simply modify the criteria. While I won't go into depth on doing this (since there are many possibilities), I will point you in the right direction. I personally use this method because it allows me to see how Microsoft is creating their reports and it also allows me to create reports that have the same look and feel as the other reports.

1. Download the .rdl file for the report that you wish to mimic. To do this, go to the MOM Reporting site and navigate to the directory where the report is stored. Click on the "Show Details" button in the upper right hand corner. Then click the "Edit" button for that report. Lastly, click the "Edit" link under the Report Definition header.
2. Save the file to any location you like.
3. Open an Existing report solution within Visual Studio or create a blank one using the report wizard.
4. Next, in the Solution Explorer (right hand side of the screen) right click on Reports and choose ADD → Existing Item
5. Next, choose the .rdl file that you downloaded. This will open the file in Visual Studio. You should now be able to see the various datasets and parameters created for this report. Although you imported the .rdl, it may not necessarily function within VS. You may need to create a data source that has the proper database connections.