



SQL Server™ 2005: Server Management Objects (SMO)

Table of Contents

SQL Server 2005: Server Management Objects (SMO).....	3
Lab Setup.....	4
Exercise 1 Creating Databases and Tables	5
Exercise 2 Backups and Verification.....	7
Exercise 3 Scripter.....	9

SQL Server 2005: Server Management Objects (SMO)

Objectives

NOTE: This lab focuses on the concepts in this module and as a result may not comply with Microsoft® security recommendations.

NOTE: The SQL Server 2005 labs are based on beta builds of the product. The intent of these labs is to provide you with a general feel of some of the planned features for the next release of SQL Server. As with all software development projects, the final version may differ from beta builds in both features and user interface. For the latest details on SQL Server 2005, please visit <http://www.microsoft.com/sql/2005/>.

Scenario

Prerequisites

Estimated Time to Complete This Lab

After completing this lab, you will be able to:

- Create Tables and Databases with Server Management Objects (SMO)
- Back up databases with SMO
- Script out database objects with SMO

This lab will show you how to use SMO to execute various tasks on SQL Server 2005 such as creating a database, creating tables, and backing up a database.

- Prior working experience with SQL Server 7.0 or SQL Server 2000
- Prior working experience with T-SQL
- Basic Microsoft Windows® networking concepts
- A basic understanding of the .NET Framework

30 Minutes

Lab Setup

Tasks	Detailed Steps
1. Log in.	1. Log in using the Administrator user account. The password is Pass@word1 .

Exercise 1

Creating Databases and Tables

Scenario

In this exercise, you will learn how to connect to the local SQL Server and how to create a database and associated tables using SMO objects.

Tasks	Detailed Steps
<p>1. Create a Windows Forms application.</p> <p>NOTE: All of the code for this exercise is contained in the Microsoft Visual Studio® solution located at C:\SQL Labs\Lab Projects\SMO\SMO.sln. You can copy and paste from these files to complete any portion of the exercise.</p>	<ol style="list-style-type: none"> 1. From the Start All Programs menu, load Microsoft Visual Studio 2005. 2. Click File New Project. 3. In the New Project dialog box, create a new Microsoft Visual Basic® project, using the Windows Application template. 4. For the project Name, enter SMO. For the project location, enter C:\SQL Labs\User Projects. Click OK to create the project.
<p>2. Add a reference to the SMO Assembly.</p>	<ol style="list-style-type: none"> 1. Select Project Add Reference. 2. Click Microsoft.SqlServer.Smo in the list, and click OK. 3. Repeat step 1 and select the Microsoft.SqlServer.ConnectionInfo assembly.
<p>3. Connect to the Local SQL Server.</p>	<ol style="list-style-type: none"> 1. Drag a Button control from the toolbox onto the form. 2. Select the new button control. 3. In the Properties window, set the (Name) property to btnConnect. 4. In the Properties window, set the Text property to Connect. 5. Double-click the new button to create an event handler for the Click event. 6. At the top of the code window, add the following code. <pre style="background-color: #f0f0f0; padding: 5px;">Imports Microsoft.SqlServer.Management.Smo</pre> 7. Inside the btnConnect click event handler add the following code. <pre style="background-color: #f0f0f0; padding: 5px;">Dim srv As New Server(Environment.MachineName) MsgBox("The AdventureWorks database was created at " & _ srv.Databases("AdventureWorks").CreateDate.ToString())</pre> <p>The first line of code creates a server object that references the local server by passing the name of the local machine to its constructor. Next, the code accesses the Databases collection of the Server and displays the creation date of the AdventureWorks database in a message box.</p> 8. Press F5 to run the Windows Forms application. 9. Click Connect. 10. Click OK in the message box.

Tasks	Detailed Steps
<p>4. Create Databases and Tables with SMO objects.</p>	<p>11. Close the window when you're done.</p> <ol style="list-style-type: none"> 1. Switch back to the design view of the form by right-clicking Form1.vb in the Solution Explorer and clicking View Designer. 2. Drag a second Button control from the toolbox onto the form. 3. Select the new button control. 4. In the Properties window, set the (Name) property to btnDBTblCreation. 5. In the Properties window, set the Text property to DB-Table. 6. Double-click the new DB-Table button to create an event handler for the button's Click event. 7. Inside the btnDBTblCreation click event handler add the following code. <pre data-bbox="553 615 1429 1119"> Dim srv As New Server(Environment.MachineName) Dim smODB As New Database(srv, "SMO_DB") smODB.Create() Dim smOTBL As New Table(smODB, "SMO_TBL") Dim smOCol1 As New Column(smOTBL, "c1", DataType.Int) smOTBL.Columns.Add(smOCol1) Dim smOCol2 As New Column(smOTBL, "c2", _ DataType.NVarChar(30)) smOTBL.Columns.Add(smOCol2) smOTBL.Create() MsgBox("Database and Table Created!") </pre> <p>The first line of code is similar to the code used before to establish a connection to the local server. Next, the code creates a new database object by calling the <code>AddNew()</code> method of the <code>Server</code> object's <code>Databases</code> collection. To create a new database on the server, the code then calls the new <code>Database</code> object's <code>Create</code> method.</p> <p>Next, the code follows a similar pattern to add a new table named "SMO_TBL" and two columns with differing data types. After that, a check constraint is created and bound to the first column. The last line of code creates the table and associated columns on the server by calling the <code>Table</code> object's <code>Create</code> method.</p> 8. Press F5 to run the Windows Forms application. 9. Click the DB-Table button. 10. Close the form when you're done.
<p>5. Verify the creation of the SMO_DB database.</p>	<ol style="list-style-type: none"> 1. From the Windows Start menu, load SQL Server Management Studio. 2. When asked to connect, make sure the server name is localhost, and that Windows Authentication is selected in the Authentication dropdown. Click Connect. 3. In the Object Explorer window, expand the Databases node, and find the new SMO_D B database. Expand the Tables node to view the new table, and drill down into the columns to view the new columns you created.

Exercise 2

Backups and Verification

Scenario

In this exercise, you will learn how to back up a SQL Server database using SMO and verify the resulting backup files.

Tasks	Detailed Steps
<p>1. Add a new button.</p> <p>NOTE: All of the code for this exercise is contained in the Visual Studio solution located at C:\SQL Labs\Lab Projects\SMO\SMO.sln. You can copy and paste from these files to complete any portion of the exercise.</p>	<ol style="list-style-type: none"> 1. Switch back to the design view of the form by right-clicking Form1.vb in the Solution Explorer and clicking View Designer. 2. Drag a third button from the toolbox onto the form. 3. Select the new button. 4. In the Properties window, set the (Name) property to btnBackup. 5. In the Properties window, set the Text property to Backup. 6. Double-click the new Backup button to create an event handler for the button click event. 7. Inside the btnBackup click event handler add the following code. <pre style="background-color: #f0f0f0; padding: 10px;"> Dim srv As New Server(Environment.MachineName) Dim backDeviceItem As New BackupDeviceItem(_ "C:\SQL Labs\User Projects\SMO_BACKUP.bak", _ DeviceType.File) Dim back As New Backup() back.Database = "AdventureWorks" back.Devices.Add(backDeviceItem) back.Action = BackupActionType.Database back.SqlBackup(srv) MsgBox("Backup Complete!") </pre> <p>The first line of code establishes a connection to the local server. Next, the code creates a backup object. The backup object encapsulates the functionality that allows you to perform database backups of SQL Server 2005 databases. Next, the code sets several properties on the database object: the name of the database to be backed up, the type of backup device to create, and the location of the backup device. The code also specifies the type of database backup action to perform. In this example, the code backs up only the database. Another option would be to perform a transaction log backup only. Finally, the code performs the backup by calling the <code>SqlBackup</code> method of the backup object and passing in the server object.</p> 8. Below the code you just added, add the following code, which verifies the backup. <pre style="background-color: #f0f0f0; padding: 10px;"> Dim rest As New Restore Dim restDeviceItem As New BackupDeviceItem(_ "C:\SQL Labs\User Projects\SMO_BACKUP.bak", _ DeviceType.File) rest.Devices.Add(restDeviceItem) </pre>

Tasks	Detailed Steps
	<pre data-bbox="548 268 1230 489"> Dim verified As Boolean = rest.SqlVerify(srv) If verified Then MsgBox("The backup was verified succesfully") Else MsgBox("The backup is corrupt") End If </pre> <p data-bbox="511 531 1414 709">To verify the backup, the code creates a Restore object. Next, the code specifies the type of backup device, and adds its location to the Restore object's Devices collection. The code then calls the Restore object's SqlVerify method, passing in the Server object. The return value from this method call, which is of type Boolean, lets the caller know whether the backup is valid or not. This is similar to executing the T-SQL statement RESTORE VERIFYONLY.</p> <p data-bbox="511 720 894 751">9. Press F5 to run the application.</p> <p data-bbox="511 762 1341 821">The following step takes several minutes, so you may want to skip to the next exercise if you are short on time.</p> <p data-bbox="511 831 1414 890">10. Click the Backup button and wait for the backup to complete. This process may take a few minutes.</p> <p data-bbox="511 900 992 930">11. When a message box appears, click OK.</p> <p data-bbox="511 940 1219 970">12. Close the application (but not the solution) when you're done.</p>

Exercise 3

Scripter

Scenario

In this exercise, you will learn how to script out a set of database objects using the Scripter object.

Tasks	Detailed Steps
<p>1. Add a reference to the SmoEnum Assembly.</p> <p>NOTE: All of the code for this exercise is contained in the Visual Studio solution located at C:\SQL Labs\Lab Projects\SMO\SMO.sln. You can copy and paste from these files to complete any portion of the exercise.</p>	<ol style="list-style-type: none"> 1. In the Solution Explorer, right-click the SMO project and click Add Reference 2. Click Microsoft.SQLServer.SmoEnum. 3. Click OK to add the reference to the project.
<p>2. Add a new button.</p>	<ol style="list-style-type: none"> 1. Switch back to the design view of the form by right-clicking Form1.vb in the Solution Explorer and clicking View Designer. 2. Drag a fourth button from the toolbox onto the form. 3. Select the new button. 4. In the Properties window, set the (Name) property to btnScripter. 5. In the Properties window, set the Text property to Scripter. 6. Double-click the new Scripter button to create an event handler for the button click event. 7. Inside the btnScripter click event handler add the following code. <pre data-bbox="553 1167 1430 1892"> Dim srv As New Server(Environment.MachineName) Dim scr As New Scripter(srv) scr.Options.FileName = _ "C:\SQL Labs\User Projects\scripter.txt" scr.Options.AppendToFile = True scr.Options.IncludeHeaders = True Dim db As Database = srv.Databases("AdventureWorks") Dim smoObjects(db.Tables.Count - 1) As SqlSmoObject Dim i As Integer = 0 For Each tbl As Table In db.Tables If Not tbl.IsSystemObject Then If tbl.Schema = "Sales" Then smoObjects(i) = tbl i += 1 End If End If Next ' Resize the array to the correct size.</pre>

Tasks	Detailed Steps
	<pre data-bbox="548 239 1015 394"> ReDim Preserve smoObjects(i - 1) scr.Script(smoObjects) MsgBox("Scripting complete!") </pre> <p data-bbox="511 405 1417 646">The first line of code establishes a connection to the local server. Next, the code creates a Scripter object, passing in the Server object. The code specifies a value for Options.FileName to indicate where the script should be saved, as well as specifying that text should be appended to the file and that descriptive headers should be included in the script. Then, the code uses SMO to retrieve an array containing information about all the tables whose schema is "Sales". The code passes this array to the Script method of the Scripter object. This results in the objects being scripted out to the specified file.</p> <ol data-bbox="511 657 1433 905" style="list-style-type: none"> 8. Run the sample project, and click the Scripter button to script the Sales tables to a text file. This process may take a few moments. 9. Open Windows Explorer. 10. Navigate to the C:\SQL Labs\User Projects folder on the local computer. 11. Double-click scripter.txt. 12. Review the DDL scripts for the tables. 13. Close the application when you're done.