

# پشتیبان گیری ، بازیابی و چک کردن پایگاه داده Sql در C# و Vb

از خواننده محترم تقاضا میشود با ارسال نظرات و پیشنهادات خود در مورد این مقاله ویا اشکالات موجود در محتوی یا متن آن به آدرس [Mehdi.aliha@gmail.com](mailto:Mehdi.aliha@gmail.com) در بهبود مقاله سهیم شود.

در این سری از مقالات ، به نحوه استفاده از SMO (Sql management Object) و انواع کارهای معمول از جمله : پشتیبان گیری، بازگرداندن، تعمیر و نگهداری ایندکسها، چک کردن وغیره که یک DBA (Data Base Admin) باید انجام دهد می پردازیم.

برخی از این عملیات ها در پایگاه داده SMO نگهداری می شوند. برای دانلود این ابزار به لینک زیر مراجعه کنید.

<http://www.sqldbtips.com/showarticle.asp?ID=29>

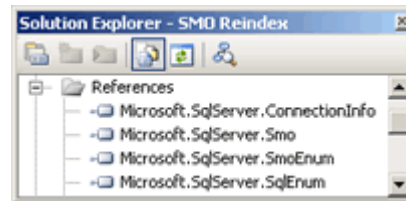
در این مقاله ما در مورد ساخت یک برنامه SMO ، وصل شدن به Sql Serve و تعیین برخی از خواص آن وهمچنین بازبازی سرور متمرکز میشویم . نمونه کدها براساس C# و VB می باشد. وهمچنین تاکید میکنم نمونه برنامه های SMO که بخشی از آنها مثالهای Sql Server Engine هستند را از لینک زیر دانلود کنید.

<http://www.codeplex.com/MSFTEngProdSamples/Release/ProjectReleases.aspx?ReleaseId=4038>

#### • ساختن یک برنامه SMO

برای استفاده از اشیاء SMO در یک برنامه NET. ما نیاز به ارجاء اسمبلی های SMO داریم ساده ترین راه برای این کار در ویژوال استودیو ۲۰۰۵ رفتن به مسیر Reference >add project و سپس از منوی اصلی اسمبلی های زیر را انتخاب و به فولدر References اضافه کنید.مانند شکل زیر

- Microsoft.SqlServer.ConnectionInfo
- Microsoft.SqlServer.Smo
- Microsoft.SqlServer.SmoEnum
- Microsoft.SqlServer.SqlEnum



برای انجام وظایف خاصی مانند مدیریت سرویس باید اسمبلی های دیگری اضافه شوند. اما اسمبلی های فوق شامل همه وظایف مورد نیاز برای انجام ابتدائی ترین مباحث می شوند. برای اطلاعات بیشتر به مرجع برنامه نویسی SMO در Books On Line رجوع شود.

اگر شما دسترسی به ویژوال استودیو ندارید. میتوانید منابع مورد نیاز برنامه کاربردی خود را در کامپایلر خط فرمان اضافه کنید. بطور مثال خطوط فرمان برای کامپایلر های C# و VB بصورت زیر می باشد.

#### [Visual Basic]

```
vbc /t:exe c:\SMOTest.vb /r:
"C:\Program Files\Microsoft SQL Server\90\SDK\Assemblies\Microsoft.SqlServer.ConnectionInfo.dll",
"C:\Program Files\Microsoft SQL Server\90\SDK\Assemblies\Microsoft.SqlServer.Smo.dll",
"C:\Program Files\Microsoft SQL Server\90\SDK\Assemblies\Microsoft.SqlServer.SmoEnum.dll",
"C:\Program Files\Microsoft SQL Server\90\SDK\Assemblies\Microsoft.SqlServer.SqlEnum.dll"
```

#### [C#]

```
csc /t:exe /out:c:\SMOTest.exe c:\SMOTest.vb /r:
"C:\Program Files\Microsoft SQL Server\90\SDK\Assemblies\Microsoft.SqlServer.ConnectionInfo.dll",
"C:\Program Files\Microsoft SQL Server\90\SDK\Assemblies\Microsoft.SqlServer.Smo.dll",
"C:\Program Files\Microsoft SQL Server\90\SDK\Assemblies\Microsoft.SqlServer.SmoEnum.dll",
"C:\Program Files\Microsoft SQL Server\90\SDK\Assemblies\Microsoft.SqlServer.SqlEnum.dll"
```

توجه داشته باشید تمام فر مانها باید در یک خط نوشته شوند. اگرچه مثالهای متعددی برای قالب بندی خطوط وجود دارد.

- اتصال به Sql

متصل شدن به Sql Server بسیار ساده است. برای مثال اگر اتصال پیش فرض در Sql Server بصورت Windows Authentication (نحوه شناسایی) تنظیم شده باشد. ما فقط نیاز به ساختن یک شی جدید از شی SMO Server داریم. همانطور که می بینید ما در مثال زیر شروع به دریافت اطلاعاتی مانند: نام سرور و نسخه نگارش سرور کردیم.

[Visual Basic]

```
Imports Microsoft.SqlServer.Management.Smo
```

```
Module SMOTest
```

```
Sub Main()
```

```
Dim svr As Server = New Server()
```

```
Console.WriteLine(svr.Name & " " & svr.Information.VersionString)
```

```
End Sub
```

```
End Module
```

[C#]

```
using System;
```

```
using Microsoft.SqlServer.Management.Smo;
```

```
namespace SMOTest
```

```
{
```

```
class Program
```

```
{
```

```
static void Main()
```

```
{
```

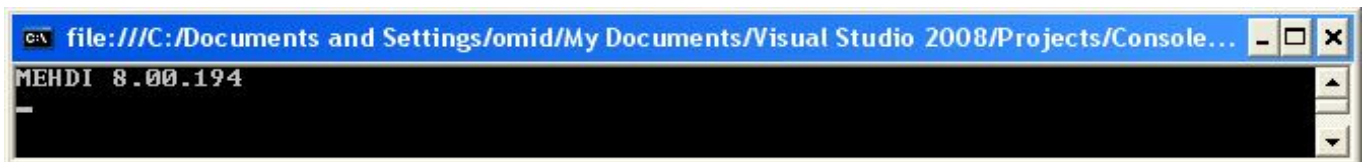
```
Server svr = new Server();
```

```
Console.WriteLine(svr.Name + " " + svr.Information.VersionString);
```

```
}
```

```
}
```

```
}
```



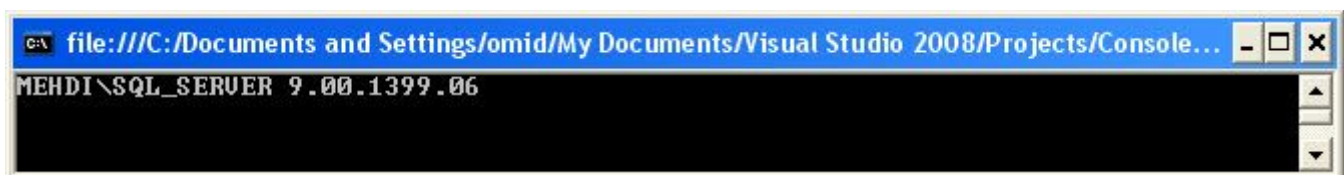
برای دسترسی به سروری با نام خاص می توان نام سرور مورد نظر را به شکل زیر به سازنده Server ارسال کرد.

```
Dim svr As Server = New Server("(local)")
```

```
Server svr = new Server(@"(local)\INSTANCE01")
```

مثال

```
Server svr = new Server(@"(local)\SQL_SERVER");
```



با این حال، اگر ما بخواهیم گزینه های پیچیده تری را برای اتصال pooling که جزئی از رفتار SMO هست را کنترل کنیم. ما می توانیم از شی ConnectionContext استفاده کنیم. این شی، فرزندی از شی Server می باشد که می توان خاصیت های آن را تعیین کرد. ویا با تعریف یک متغیر محلی به عنوان یک شی ServerConnection و خاصیت های آن را تعیین و به متد سازنده Server ارسال کنیم.

مثال زیر خاصیت هایی مانند: Authentication mode و Password و UserName را تنظیم می کند.

[Visual Basic]

```
Imports Microsoft.SqlServer.Management.Smo
Imports Microsoft.SqlServer.Management.Common
```

Module SMOTest

Sub Main()

```
Dim svr As Server = New Server()
svr.ConnectionContext.LoginSecure = false
svr.ConnectionContext.Login = "username"
svr.ConnectionContext.Password = "password"
Console.WriteLine(svr.Name & " " & svr.Information.VersionString)
```

End Sub

End Module

[C#]

```
using System;
using Microsoft.SqlServer.Management.Smo;
using Microsoft.SqlServer.Management.Common;
```

namespace SMOTest

```
{
    class Program
    {
        static void Main()
        {
            ServerConnection conn = new ServerConnection();
            conn.LoginSecure = false;
            conn.Login = "username";
            conn.Password = "password";
            Server svr = new Server(conn);
            Console.WriteLine(svr.Name + " " + svr.Information.VersionString);
        }
    }
}
```

• رفتار پیش فرض- اتصال مخلوط

در رفتار پیش فرض، SMO با استفاده از اتصال مخلوط یا Pooling، در خواست برقراری یا رها سازی اتصالات رو انجام می دهد. هرچند این رفتار می تواند توسط AutoDisconnectMode که از خاصیت های ConnectionContext می باشد. به AutoDisconnectMode.NoAutoDisconnect یا به NonPooledConnection تغییر یابد.

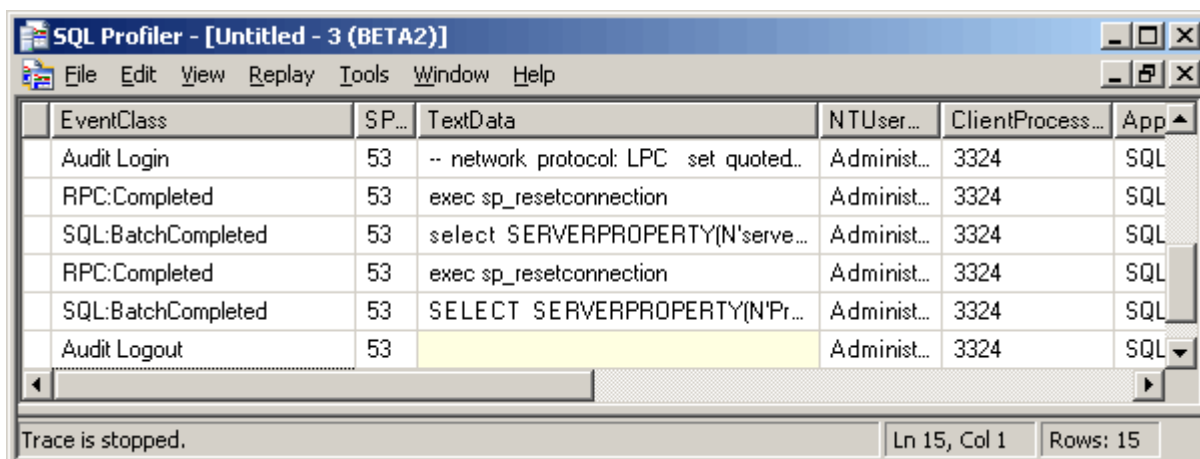
اگر AutoDisconnectMode به NoAutoDisconnect تنظیم شده باشد و اتصال نیز صریح باشد، تا زمان قطع شدن اتصال Pool اتصال قابل بازگشت نیست. چون اتصال از نوع مخلوط است. اما اگر خاصیت NonPooledConnection به True تغییر یابد ماصریحاً در خواست اتصال غیر مخلوط را کرده ایم. با استفاده از Profiler در Sql Server می توان رفتار های بر نامه SMO را مشاهده نمود.

مثال

رفتار پیش فرض- اتصال مخلوط

```
Dim svr As Server = New Server()
```

```
Console.WriteLine(svr.Name & " " & svr.Information.VersionString)
```



EventClass	SP...	TextData	NTUser...	ClientProcess...	App
Audit Login	53	-- network protocol: LPC set quoted..	Administ...	3324	SQL
RPC:Completed	53	exec sp_resetconnection	Administ...	3324	SQL
SQL:BatchCompleted	53	select SERVERPROPERTY(N'serve...	Administ...	3324	SQL
RPC:Completed	53	exec sp_resetconnection	Administ...	3324	SQL
SQL:BatchCompleted	53	SELECT SERVERPROPERTY(N'Pr...	Administ...	3324	SQL
Audit Logout	53		Administ...	3324	SQL

Trace is stopped. Ln 15, Col 1 Rows: 15

```
Dim svr As Server = New Server()
svr.ConnectionContext.NonPooledConnection = True
svr.ConnectionContext.Connect()
Console.WriteLine(svr.Name & " " & svr.Information.VersionString)
svr.ConnectionContext.Disconnect()
```

EventClass	SP...	TextData	NTUser...	ClientP
Audit Login	53	-- network protocol: LPC set quoted_identifier on...	Administ...	3252
SQL:BatchCompleted	53	select SERVERPROPERTY(N'servername')	Administ...	3252
SQL:BatchCompleted	53	SELECT SERVERPROPERTY(N'ProductVersion')...	Administ...	3252
Audit Logout	53		Administ...	3252

Trace is stopped. Ln 13, Col 1 Rows: 13

### • پشتیبان گیری DataBase

برای تهیه یک نسخه پشتیبان مانیا به دوشی داریم. ۱- شی Server ۲- شی Backup

در ساده ترین شکل مانیا به یک شی Backup داریم. که قبل از فراخوانی متد SqlBackup باید خواص شی Backup را تنظیم و شی Server را به آن ارسال کنیم. در مثال زیر یک پشتیبانی کامل از پایگاه داده SMO در مسیر C:\SMOTest.bak گرفته شده است.

[Visual Basic]

```
Imports Microsoft.SqlServer.Management.Smo
```

```
Module SMOTest
```

```
Sub Main()
```

```
Dim svr As Server = New Server()
Dim bkp As Backup = New Backup()
bkp.Devices.AddDevice("C:\SMOTest.bak", DeviceType.File)
bkp.Database = "SMO"
bkp.Action = BackupActionType.Database
bkp.Initialize = True
bkp.PercentCompleteNotification = 10
AddHandler bkp.PercentComplete, AddressOf ProgressEventHandler
bkp.SqlBackup(svr)
```

```
End Sub
```

```
Private Sub ProgressEventHandler(ByVal sender As Object, _
ByVal e As PercentCompleteEventArgs)
Console.WriteLine(e.Percent.ToString + "% backed up")
```

```
End Sub
```

```
End Module
```

[C#]

```
using Microsoft.SqlServer.Management.Smo;

namespace SMOTest
{
    class Program
    {
        static void Main()
        {
            Server svr = new Server();
            Backup bkp = new Backup();
            bkp.Devices.AddDevice(@"C:\SMOTest.bak", DeviceType.File);
            bkp.Database = "SMO";
            bkp.Action = BackupActionType.Database;
            bkp.Initialize = true;
            bkp.PercentCompleteNotification = 10;
            bkp.PercentComplete += new
                PercentCompleteEventHandler(bkp_PercentComplete);
            bkp.SqlBackup(svr);
        }

        static void bkp_PercentComplete(object sender, PercentCompleteEventArgs e)
        {
            Console.WriteLine(e.Percent.ToString() + "% backed up");
        }
    }
}
```

در این تمرین از یک Event Handler در متد bkp\_PercentComplete استفاده شده که وضعیت پشتیبان گیری را نمایش می دهد.

خصوصیات زیادی برای تنظیم شی Backup وجود دارد. برای دیدن لیست تمام این خصوصیات می توانید به [books on line](#) مراجعه کنید. (اگر شما می خواهید کتاب های رایگان 2005 Sql را نصب کنید به لینک زیر مراجعه کنید).

<http://www.microsoft.com/downloads/details.aspx?familyid=BE6A2C5D-00DF-4220-B133-29C1E0B6585F&displaylang=en>

- پشتیبان گیری بصورت Filegroup

در مثال زیر از WorkOrderGroup که در پایگاه داده Adventureworks قرار دارد بصورت filegroup پشتیبان گیری شده است. (اسکرپت تبدیل پایگاه داده مجازی Adventureworks به چندین filegroup را می توان در مثالهای SQL Server Engine پیدا کرد مسیر پیش فرض قرار گیری مثالها (C:\Program Files\Microsoft SQL Server\90\Samples\Engine\Administration\FileGroups\Scripts) می باشد).

Filegroup ها دارای Backup های خاصی هستند که بصورت اشیاء Backup به مجموعه DatabaseFileGroups اضافه میشوند. به منظور تهیه Backup های خاص می توان آنها را به مجموعه DatabaseFileGroup اضافه نمود.

[Visual Basic]

```
Imports Microsoft.SqlServer.Management.Smo

Module SMOTest
    Sub Main()
        Dim svr As Server = New Server()
        Dim bkp As Backup = New Backup()
        bkp.Database = "AdventureWorks"
        bkp.Action = BackupActionType.Files
        bkp.DatabaseFileGroups.Add("WorkOrderGroup")
        bkp.Devices.AddDevice("C:\AWFGTest.bak", DeviceType.File)
        bkp.SqlBackup(svr)
    End Sub
End Module
```

[C#]

```
using Microsoft.SqlServer.Management.Smo;

namespace SMOTest
{
    class Program
    {
        static void Main()
        {
            Server svr = new Server();
            Backup bkp = new Backup();
            bkp.Database = "AdventureWorks";
            bkp.Action = BackupActionType.Files;
            bkp.DatabaseFileGroups.Add("WorkOrderGroup");
            bkp.Devices.AddDevice(@"C:\AWFGTest.bak", DeviceType.File);
            bkp.SqlBackup(svr);
        }
    }
}
```

• پشتیبان گیری بصورت Differential یا دیفرانسیل

تهیه یک نسخه پشتیبان بصورت Differential بسیار ساده است. فقط کافیست خاصیت Incremental از شی Backup را به True تنظیم کنیم.

```
bkp.Incremental = True
```

پشتیبان گیری بصورت Log

پشتیبان گیری Log بسیار ساده است. فقط کافیست خاصیت Action از شی Backup را به BackupActionType.Log تنظیم کنیم.

[Visual Basic]

```
Imports Microsoft.SqlServer.Management.Smo

Module SMOTest
    Sub Main()
        Dim svr As Server = New Server()
        Dim bkp As Backup = New Backup()
        bkp.Action = BackupActionType.Log
        bkp.Database = "SMO"
        bkp.Devices.AddDevice("c:\SMOTest.trn", DeviceType.File)
        bkp.SqlBackup(svr)
    End Sub
End Module
```

[C#]

```
using Microsoft.SqlServer.Management.Smo;

namespace SMOTest
{
    class Program
    {
        static void Main()
        {
            Server svr = new Server();
            Backup bkp = new Backup();
            bkp.Action = BackupActionType.Log;
            bkp.Database = "SMO";
            bkp.Devices.AddDevice(@"c:\SMOTest.trn", DeviceType.File);
            bkp.SqlBackup(svr);
        }
    }
}
```

برای بازیابی یک DataBase مانیا به دوشی داریم. ۱- شی Server ۲- شی Restore

در ساده ترین شکل مانیا به یک شی Restore داریم. که قبل از فراخوانی متد SqlRestore باید خواص شی Restore را تنظیم و شی Server را به آن ارسال کنیم. در مثال زیر یک بازیابی کامل از پایگاه داده SMO در مسیر C:\SMOTest.bak جایگزین پایگاه داده فعلی می شود.

[Visual Basic]

```
Imports Microsoft.SqlServer.Management.Smo
```

```
Module SMOTest
```

```
Sub Main()
```

```
Dim svr As Server = New Server()
Dim res As Restore = New Restore()
res.Devices.AddDevice("C:\SMOTest.bak", DeviceType.File)
res.Database = "SMO"
res.ReplaceDatabase = True
res.PercentCompleteNotification = 10
AddHandler res.PercentComplete, AddressOf ProgressEventHandler
res.SqlRestore(svr)
```

```
End Sub
```

```
Private Sub ProgressEventHandler(ByVal sender As Object, _
ByVal e As PercentCompleteEventArgs)
Console.WriteLine(e.Percent.ToString + "% restored")
```

```
End Sub
```

```
End Module
```

[C#]

```
using System;
```

```
using Microsoft.SqlServer.Management.Smo;
```

```
namespace SMOTest
```

```
{
    class Program
    {
        static void Main()
        {
            Server svr = new Server();
            Restore res = new Restore();
            res.Database = "SMO";
            res.Action = RestoreActionType.Database;
            res.Devices.AddDevice(@"C:\SMOTest.bak", DeviceType.File);
            res.PercentCompleteNotification = 10;
            res.ReplaceDatabase = true;
            res.PercentComplete += new
                PercentCompleteEventHandler(ProgressEventHandler);
            res.SqlRestore(svr);
        }

        static void ProgressEventHandler(object sender, PercentCompleteEventArgs e)
        {
            Console.WriteLine(e.Percent.ToString() + "% restored");
        }
    }
}
```

در این تمرین از یک Event Handler در متد ProgressEventHandler استفاده شده که وضعیت پشتیبان گیری را نمایش می دهد.

خصوصیات زیادی برای تنظیم شی Restore وجود دارد. برای دیدن لیست تمام این خصوصیات می توانید به [books on line](#) مراجعه کنید. (اگر شما می خواهید کتاب های رایگان Sql 2005 را نصب کنید به لینک زیر مراجعه کنید).

<http://www.microsoft.com/downloads/details.aspx?familyid=BE6A2C5D-00DF-4220-B133-29C1E0B6585F&displaylang=en>



با استفاده از معادل دستورات T-SQL WITH MOVE برای بازیابی و همچنین تنظیم خاصیت‌های RelocateFiles از شی Restore

و تعیین پارامترهای شی RelocateFile به کمک SMO این عمل را انجام می‌دهیم. در مثال زیر ما یک کپی از پایگاه داده SOM را به همراه داده‌ها و فایل‌های Log در پایگاه داده SMO2 بازیابی کردیم.

در سازنده RelocateFile از دو پارامتر استفاده شده: ۱- نام منطقی فایل ۲- نام فیزیکی فایل. که نقش آن فراهم کردن محلی برای انتقال فایل‌ها در زمان بازیابی است.

[Visual Basic]

```
Imports Microsoft.SqlServer.Management.Smo
```

```
Module SMOTest
```

```
Sub Main()
```

```
Dim svr As Server = New Server()
```

```
Dim res As Restore = New Restore()
```

```
res.Devices.AddDevice("C:\SMOTest.bak", DeviceType.File)
```

```
res.Database = "SMO2"
```

```
res.NoRecovery = False
```

```
res.ReplaceDatabase = True
```

```
res.RelocateFiles.Add(New RelocateFile("SMO", "c:\SMO2.mdf"))
```

```
res.RelocateFiles.Add(New RelocateFile("SMO_Log", "c:\SMO2.ldf"))
```

```
res.SqlRestore(svr)
```

```
End Sub
```

```
End Module
```

[C#]

```
using Microsoft.SqlServer.Management.Smo;
```

```
namespace SMOTest
```

```
{
```

```
class Program
```

```
{
```

```
static void Main()
```

```
{
```

```
Server svr = new Server();
```

```
Restore res = new Restore();
```

```
res.Database = "SMO2";
```

```
res.Action = RestoreActionType.Database;
```

```
res.Devices.AddDevice(@"C:\SMOTest.bak", DeviceType.File);
```

```
res.ReplaceDatabase = true;
```

```
res.RelocateFiles.Add(new RelocateFile("SMO", @"c:\SMO2.mdf"));
```

```
res.RelocateFiles.Add(new RelocateFile("SMO_Log", @"c:\SMO2.ldf"));
```

```
res.SqlRestore(svr);
```

```
}
```

```
}
```

```
}
```

- اطلاعاتی در مورد فایل‌های پشتیبان

تعدادی متد در شی Restore وجود دارد که با استفاده از هر کدام می‌تواند اطلاعاتی در مورد درایو Backup و فایل‌های Backup نشان داد. که این متد ها عبارتند از : ۱- ReadBackupHeader ۲- ReadFileList ۳- ReadMediaHeader .

در مثال زیر ما با استفاده از متد ReadFileList لیستی از نام فایل‌های منطقی در درایو c:\SMOTest.bak را در کنسول نمایش می‌دهیم.

[Visual Basic]

```
Imports Microsoft.SqlServer.Management.Smo
```

```
Imports System.Data
```

```
Module SMOTest
```

```
Sub Main()
```

```
Dim svr As Server = New Server()
```

```
Dim res As Restore = New Restore()
```

```
Dim dt As DataTable
```

```
Dim foundrows As DataRow()
```

```
res.Devices.AddDevice("C:\SMOTest.bak", DeviceType.File)
```

```
dt = res.ReadFileList(svr)
```

```
foundrows = dt.Select(Nothing)
```

```
For Each dr As DataRow In foundrows
```

```
Console.WriteLine(dr["LogicalName"].ToString())
```

```
Next
```

```
End Sub
```

```
End Module
```

[C#]

```
using System;
```

```
using System.Data;
```

```
using Microsoft.SqlServer.Management.Smo;
```

```
namespace SMOTest
```

```
{
```

```
class Program
```

```
{
```

```
static void Main()
```

```
{
```

```
Server svr = new Server();
```

```
Restore res = new Restore();
```

```
DataTable dt;
```

```
DataRow[] foundrows;
```

```
res.Devices.AddDevice(@"C:\SMOTest.bak", DeviceType.File);
```

```
dt = res.ReadFileList(svr);
```

```
foundrows = dt.Select();
```

```
foreach (DataRow r in foundrows)
```

```
{
```

```
Console.WriteLine(r["LogicalName"].ToString());
```

```
}
```

```
}
```

```
}
```

از آنجا که همه این متد ها شی از نوع DataTable بر می‌گردانند . می‌توان به راحتی به همه ستون های استخراج شده دسترسی داشت. مثال زیر، مجموعه ای از ستونهای استخراج شده از شی DataTable را نمایش می‌دهد.

```
[Visual Basic]
Imports Microsoft.SqlServer.Management.Smo
Imports System.Data
```

```
Module SMOTest
    Sub Main()
        Dim svr As Server = New Server()
        Dim res As Restore = New Restore()
        Dim dt As DataTable
        Dim foundrows As DataRow()

        res.Devices.AddDevice("C:\SMOTest.bak", DeviceType.File)
        dt = res.ReadFileList(svr)

        foundrows = dt.Select(Nothing)

        For Each col As DataColumn In dt.Columns
            Console.WriteLine(col.ColumnName + vbTab)
        Next

        Console.WriteLine()

        For Each dr As DataRow In foundrows
            For Each col As DataColumn In dt.Columns
                Console.WriteLine(dr(col.ColumnName).ToString() + vbTab)
            Next
            Console.WriteLine()
        Next
    End Sub
End Module
```

• بازبازی Log

بازبازی Log بسیار ساده است . فقط کافی است خاصیت Action از شی Restore را به RestoreActionType.Log تنظیم کنیم . خاصیت های دیگر برای پشتیبان گیری Log وجود دارند مثل ToPointInTime که با استفاده از آن می توان عمل بازبازی را به نقطه خاصی از زمان معطوف کرد.

```
[Visual Basic]
Imports Microsoft.SqlServer.Management.Smo
```

```
Module SMOTest
    Sub Main()
        Dim svr As Server = New Server()
        Dim res As Restore = New Restore()
        res.Devices.AddDevice("C:\SMOTest.trn", DeviceType.File)
        res.Database = "SMO"
        res.NoRecovery = False
        res.Action = RestoreActionType.Log
        res.SqlRestore(svr)
    End Sub
End Module
```

```
[C#]
```

```
using System;
using Microsoft.SqlServer.Management.Smo;

namespace SMOTest
{
    class Program
    {
        static void Main()
        {
            Server svr = new Server();
            Restore res = new Restore();
            res.Database = "SMO";
            res.Action = RestoreActionType.Log;
            res.Devices.AddDevice(@"C:\SMOTest.trn", DeviceType.File);
            res.NoRecovery = false;
            res.SqlRestore(svr);
        }
    }
}
```

• چک کردن درستی DataBase

برای چک کردن درستی DataBase از متد های SMO استفاده می شود.

در جدول زیر متد های SMO همراه با دستورات معادل TSQL آن را می بینید.

SMO Method	TSQL Command
CheckAllocations(RepairType.None) CheckAllocationsDataOnly() CheckCatalog() CheckTables(RepairType.None) CheckTablesDataOnly()	DBCC CHECKALLOC WITH NO_INFOMSGS DBCC CHECKALLOC('databasename', NOINDEX) DBCC CHECKCATALOG DBCC CHECKDB WITH NO_INFOMSGS DBCC CHECKDB('databasename', NOINDEX)

همه متدها در جدول بالا یک مقدار StringCollection برمی گردانند. ما میتوانیم با تکرار این عمل نتایج برگشتی را چک کنیم تا از درستی بانک مطمئن شویم.

[Visual Basic]

```
Imports Microsoft.SqlServer.Management.Smo
Imports System.Collections.Specialized
```

Module SMOTest

Sub Main()

```
Dim svr As Server = New Server()
Dim sc As StringCollection
Dim db As Database
db = svr.Databases("AdventureWorks")
```

```
sc = db.CheckTables(RepairType.None)
```

```
For c As Integer = 0 To (sc.Count - 1)
    If sc(c).Length > 0 Then
        Console.WriteLine(sc(c))
    End If
Next
```

End Sub

End Module

[C#]

```
using System;
using Microsoft.SqlServer.Management.Smo;
using System.Collections.Specialized;
```

namespace SMOTest

```
{
    class Program
    {
        static void Main()
        {
            Server svr = new Server();
            Database db;
            db = svr.Databases["AdventureWorks"];

            StringCollection sc;
            sc = db.CheckCatalog();

            for (int i = 0; i < sc.Count; i++)
            {
                if (sc[i].Length > 0)
                    Console.WriteLine(sc[i]);
            }
        }
    }
}
```

برای شناسایی یک Backup مانیز به دوشی داریم .۱- شی Server ۲- شی Restore

در ساده ترین شکل مانیز به یک شی Restore داریم .که قبل از فراخوانی متد SqlVerify باید خواص شی Restore را تنظیم و شی Server را به آن ارسال کنیم. در مثال زیر یک شناسایی از یک Backup پایگاه داده در مسیر C:\SMOTest.bak انجام شده. در این مثال متد SqlVerify سربارگذاری گردیده. بطوری که در صورت بر گشت مقدار False از متد SqlVerify خطای Verify آن نمایش داده میشود.

[Visual Basic]

```
Imports Microsoft.SqlServer.Management.Smo
```

```
Module SMOTest
```

```
Sub Main()
```

```
Dim svr As Server = New Server()
```

```
Dim res As Restore = New Restore()
```

```
Dim verified As Boolean
```

```
Dim errmsg As String = ""
```

```
res.Devices.AddDevice("C:\SMOTest.bak", DeviceType.File)
```

```
verified = res.SqlVerify(svr, errmsg)
```

```
Console.WriteLine("verified = " & verified.ToString())
```

```
If verified = False Then
```

```
Console.WriteLine(errormsg)
```

```
End If
```

```
End Sub
```

```
End Module
```

[C#]

```
using System;
```

```
using Microsoft.SqlServer.Management.Smo;
```

```
namespace SMOTest
```

```
{
```

```
class Program
```

```
{
```

```
static void Main()
```

```
{
```

```
Server svr = new Server();
```

```
Restore res = new Restore();
```

```
Boolean verified;
```

```
String errmsg;
```

```
res.Devices.AddDevice(@"C:\SMOTest.bak", DeviceType.File);
```

```
verified = res.SqlVerify(svr, out errmsg);
```

```
Console.WriteLine("verified = " + verified.ToString());
```

```
if (verified == false)
```

```
Console.WriteLine(errormsg);
```

```
}
```

```
}
```

```
}
```

[Mehdi.aliha@gmail.com](mailto:Mehdi.aliha@gmail.com)