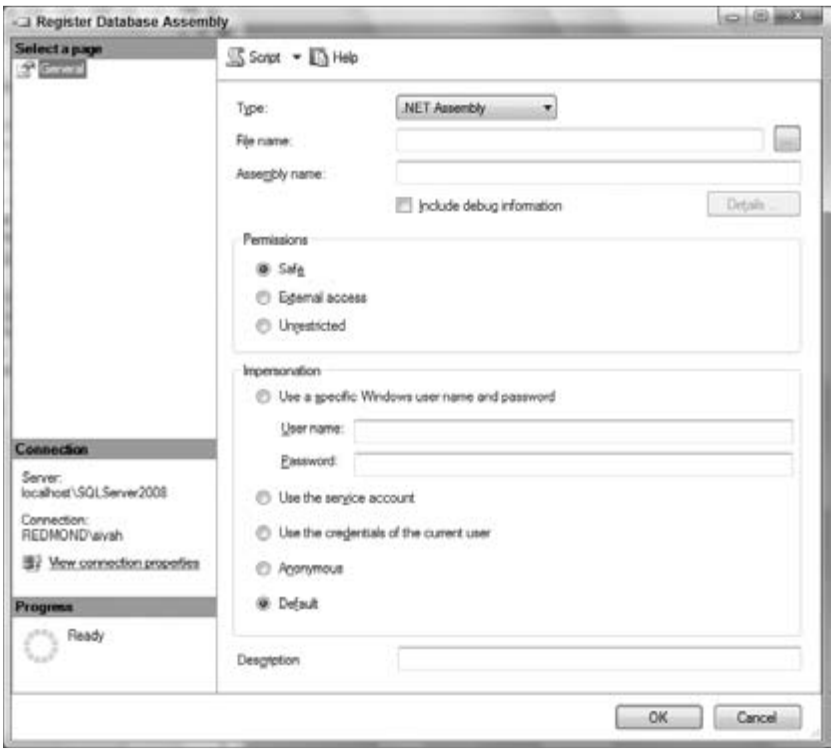
Лекция 15

Тема « Managing Assemblies »

Managing Assemblies

Assemblies, also called stored procedures, help you in performing specific tasks on the Analysis Services database or across the server. For example, Analysis Services has four assemblies installed that provide you with the functionality of calling Excel or VBA functions within your MDX queries. The System Assembly is used for operations such as Backup or Restore in retrieving information such as folders containing Analysis Services backup files, as well as supporting data mining algorithm requests. Analysis Services 2008 supports two types of assemblies: COM user - defined functions (UDFs) and .NET assemblies. COM UDFs are primarily supported for backwards compatibility with Analysis Services 2000. You learn about .NET and COM assemblies and how to build and deploy them in Chapter 11 . In this section you learn about managing assemblies on your Analysis Services instance. Assemblies can be added only by Analysis Services administrators. You need to make sure your instance of Analysis Services is safe and secure irrespective of the operations done by the stored procedures. Security is always a concern, and you do not want any assemblies to bring down the server. Because hackers try to hack servers, most software products now are built to be secure by default. The administrator needs to enable certain components and options to make them available to users. By default, Analysis Services does not allow execution of stored procedures. The administrator first needs to enable the server property Feature\ComUdfEnabled to true (value of 1 in the Analysis Services config file) for enabling COM UDFs. This is accomplished using the Analysis Server Properties dialog.
The key to managing assemblies is to understand the nature of the assembly and setting appropriate properties while adding assemblies to your Analysis Services server. Figure 7 - 27 shows the dialog used to add assemblies to the server or to a specific database. This dialog can be launched by right - clicking the Assemblies folder under a specific database and choosing New Assembly.



Analysis Services supports two types of assemblies: COM and .NET CLR assemblies. Once you specify the type and name of the assemblies in the Register Assembly dialog, you need to specify the security information for these assemblies. Two parameters control the security of these stored procedures: Impersonation and Permissions. Permissions allow you to define the scope of access for the assembly,

such as accessing the file system, accessing the network, and accessing unmanaged code. There are three different values for permissions. They are:

Safe: The most secure of the three permissions. When the Safe permission set is specified for an assembly, it means that the assembly is intended only for computation and the assembly cannot access any protected resource. It guarantees protection against information leaks and elevation attacks by malicious code.

External access: This permission value allows access to external resources by the assembly without compromising reliability, but does not offer any specific security guarantees. You can use this if you as the DBA trust the programmer ' s ability to write good code and if there is a need to access external resources such as data from an external file.

Unrestricted: This set value is primarily intended for people who have a very good understanding of programming on servers and need access to all resources. This permission set does not guarantee any code security or reliability. Unrestricted access should only be allowed to assemblies that have been written by users who absolutely need access to external resources and have a very good understanding of all security issues, such as denial of service attacks and information leakage, and are able to handle all these within the stored procedures. We recommend you use this option only when it is absolutely essential and you have full confidence in the programming abilities of the developer who has developed the assembly.

All COM DLLs will have the Permissions parameter set to Unrestricted. The Impersonation parameter allows you to specify the account under which the stored procedure will be executed. There are five different values for Impersonation:

Default: The Default value allows you to execute the stored procedure under a secure mode with the minimum privileges. If the assembly is of type COM the default value is " Use the credentials of the current user. " For a .NET assembly, the default value depends on the permission set defined. If the permission set is Safe, the Impersonation mode will be Impersonate Service Account, but if the permission set is External Access or Unrestricted, the Impersonation mode will be Impersonate Current User.

Anonymous: If you want the stored procedure to be executed as an anonymous user, you need to select Impersonate Anonymous. You will have limited access when the stored procedure is executed under this setting.

Use the credentials of the current user: This impersonation mode is typically used when you want the stored procedure to be executed with the user ' s credentials. This is a safe option to select. If the stored procedure accesses external resources and the current user executing the stored procedure does not have permissions, execution of the stored procedure will not cause any ill effects. A use of this impersonation mode is to define dynamic data security where the current user ' s credential is needed to access external resources.

Use the service account: If you choose to use the service account, whenever the stored procedure is executed it will be executed under the credentials of service startup account for Analysis Services. An example of a stored procedure that would need this impersonation mode is an AMO stored procedure that does management operations on the server.

Use a specific Windows username and password: If your business needs a stored procedure to always be executed in the context of a specific user, you need to choose this option. You need to specify a Windows account name and password for this impersonation mode. A typical service to retrieve data with this account and utilize that value within the stored procedure for computation. If you choose this option, you will need to make sure you update the password on the account when there is a password change.

We recommend that COM assemblies use the credentials of the current user impersonation, whereas for .NET CLR assemblies you should use the appropriate impersonation mode based on your customer scenario. As an administrator of Analysis Services, you need to choose the impersonation and permission setting that suits your business needs and does not compromise the security of your Analysis Services instance.

When you register an assembly with a specific Analysis Services database or for the server using the Register Assembly dialog, AMO will be used to set up the correct properties. This, in turn, sends a Create command to the Analysis Services instance as shown here:

```
< Create AllowOverwrite="true" xmlns="http://schemas.microsoft.com/
analysisservices/2003/engine" >
< ParentObject >
< DatabaseID > AnalysisServices2008Tutorial < /DatabaseID >
< /ParentObject >
< ObjectDefinition >
< Assembly xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ddl2="http://schemas.microsoft.com/analysisservices/2003/
```

engine/2" xmlns:ddl2_2="http://schemas.microsoft.com/analysisservices/
2003/engine/2/2" xmlns:ddl100_100="http://schemas.microsoft.com/
analysisservices/2008/engine/100/100" xsi:type="ClrAssembly" >
< ID > AmoSproc < /ID >
< Name > AmoSproc < /Name >
< Description / >
< ImpersonationInfo >
< ImpersonationMode > Default < /ImpersonationMode >
< /ImpersonationInfo >
< Files >
< File >
< Name > AmoSproc.dll < /Name >
< Type > Main < /Type >
< Data >
< Block > ----------Content about the stored procedure------
< /Block >
< Block > ----------Content about the stored procedure------
< /Block >
< Block > ----------Content about the stored procedure------
< /Block >
< Block > ----------Content about the stored procedure------
< /Block >
< /Data >
< /File >
< /Files >
< PermissionSet > Safe < /PermissionSet >
< /Assembly >
< /ObjectDefinition >
< /Create >

The information within the BLOCK tag is a large amount of text content, which for illustration purposes has been restricted to a single line. This text within the BLOCK tag is the assembly to be registered that will be stored within the Analysis Services instance. When queries use functions within the assembly, Analysis Services loads the assembly within the same process and executes the CLR assembly with appropriate parameter passing. The results from the assembly are appropriately passed back to Analysis Services for further evaluation of a query.

Backup and Restore

Backup is an operation that is part of every individual ' s life. If you have an important document, you make a photocopy as a backup. Similarly, backup is an extremely critical operation for any data warehouse. There are several reasons why you should periodically back up your Analysis Services database. One reason is for disaster recovery; another is for auditing purposes. Irrespective of purpose, it is always a good idea to back up your database on a periodic basis. You can back up databases on your Analysis Services instance through SSMS. Follow these steps to back up the AnalysisServices2008Tutorial database:

1. Connect to the Analysis Services instance using SSMS.
2. Navigate to the database AnalysisServices2008Tutorial in the Object Explorer window.
3. Right - click the database and select Back Up.

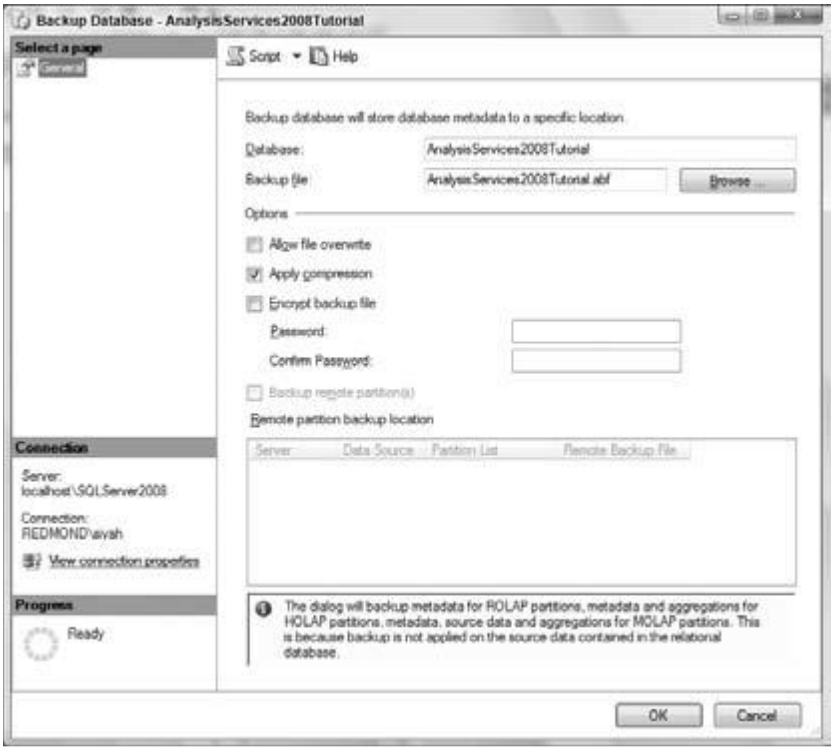
You will see the Backup dialog shown in Figure 7 - 28 . By default the dialog chooses the database name as the backup name. By default the backup file will be created in the Backup folder of your Analysis Services installation. If you want the backup to be stored in a location on a different drive or directory, you first need to change the Analysis Services server property AllowedBrowsingFolder by adding the appropriate directory. You can then choose the folder by clicking Browse in the Backup Database dialog.

You have the option to encrypt the database and specify a password. You ' ll need that password to restore the database. If you have remote partitions in the database, you have the option of specifying the backup location for each remote partition. Backup of these partitions is done on respective Analysis Services instances on that machine.

4. Disable the option to Encrypt the backup file.
5. Select the option " Allow file overwrite " to overwrite any existing backup files with the same name.
6. Choose the default backup file name and click OK.

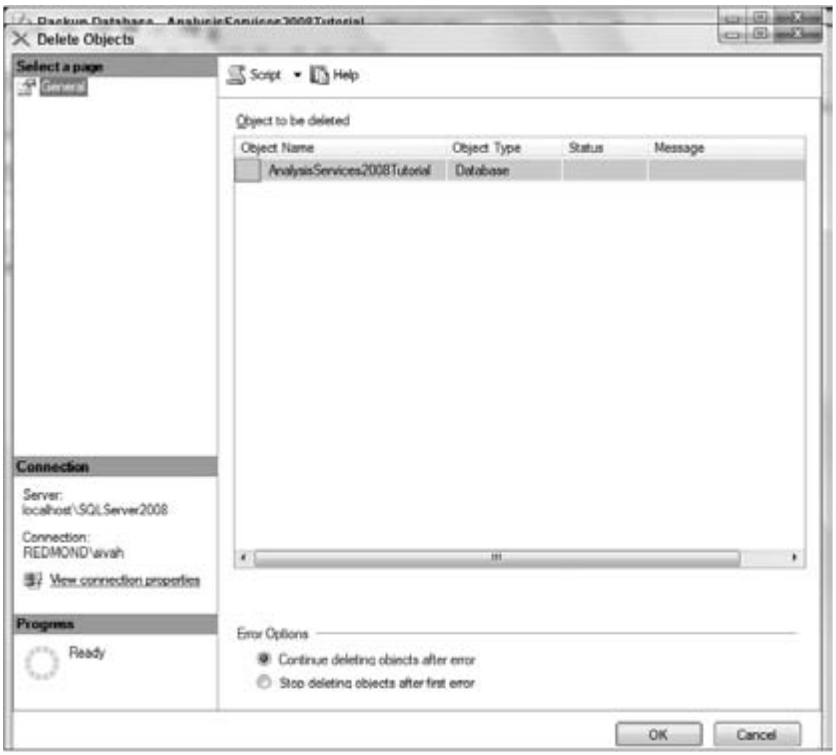
The following command is sent to the Analysis Services instance by SSMS to back up the database AnalysisServices2008Tutorial:

```
< Backup xmlns="http://schemas.microsoft.com/analysisservices/2003/engine" >
< Object >
< DatabaseID > AnalysisServices2008Tutorial < /DatabaseID >
< /Object >
< File > AnalysisServices2008Tutorial.abf < /File >
< AllowOverwrite > true < /AllowOverwrite >
< /Backup >
```

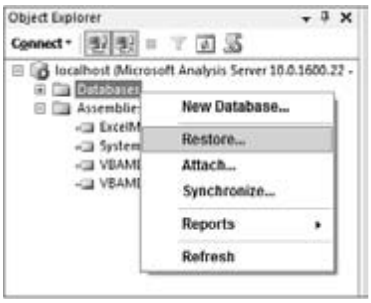


If you have specified a password, an Analysis Services 2008 backup file with the extension .abf will be created in the Backup folder. Backing up Analysis Services 2005 databases of sizes greater than 10GB used to take a long time. Analysis Services 2008 has made specific enhancements to backup performance intended to result in shorter backup times for databases of any size. Analysis Services 2008 also allows you to back up multiple databases at the same time. Through the SQL Server Management Studio you can launch the backup command from multiple databases and run backups in parallel. Alternatively, you can create a DDL that will execute backup of multiple databases within the same command. Whenever you want to restore an Analysis Services database for which you have a backup, you can do so using the Restore Database dialog. Follow these steps to restore the AnalysisServices2008Tutorial backup:

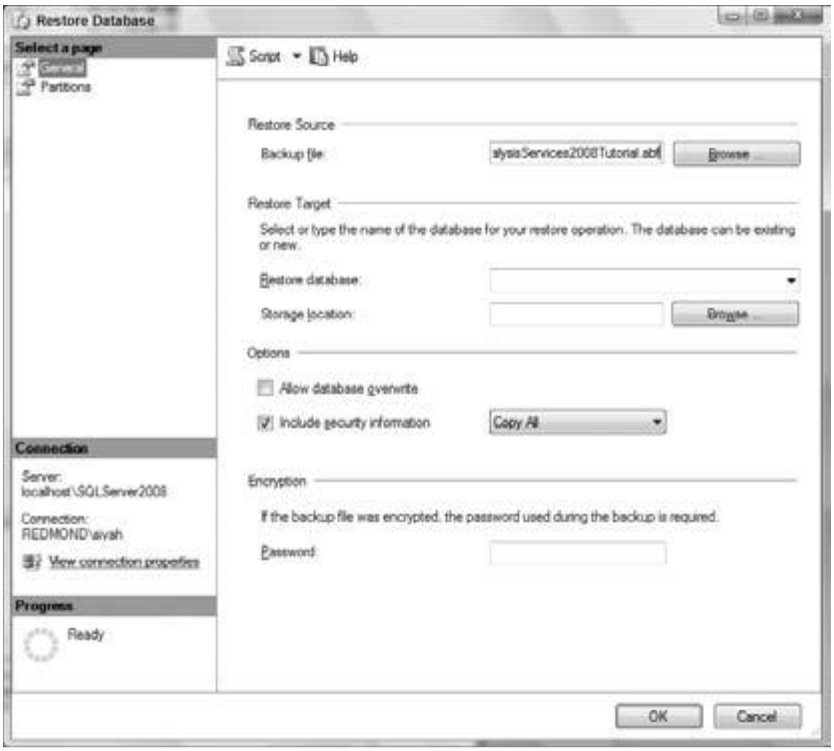
1. In SSMS Object Explorer, right - click the AnalysisServices2008Tutorial and select Delete.
2. In the Delete Objects dialog shown in Figure 7 - 29 , click OK.



3. In the SSMS Object Explorer, right - click the Databases folder as shown in Figure 7 - 30 and select Restore.



4. In the Restore Database dialog (see Figure 7 - 31) click the Browse button next to Backup File.



5. In the Locate Database Files dialog, navigate to the Backup folder and select the AnalysisServices2008Tutorial.abf file as shown in Figure 7 - 32 and click OK.

6. Type **AnalysisServices2008Tutorial** in the combo box next to Restore Database and click OK. SSMS now sends the following XMLA command to restore the database on your Analysis Services

instance:

```
< Restore xmlns="http://schemas.microsoft.com/analysisservices/2003/engine" >
< File > C:\Program Files\Microsoft SQL Server\MSAS10.MSSQLSERVER\OLAP\Backup\
AnalysisServices2008Tutorial.abf < /File >
< DatabaseName > AnalysisServices2008Tutorial < /DatabaseName >
< /Restore >
```

If you refresh the list of databases on your Analysis Services instance, you should now see the AnalysisServices2008Tutorial database in the SSMS Object Explorer. If a database with the same name and ID exists on your Analysis Services instance, you can restore the newer database by clicking the Allow Database Overwrite checkbox in the Restore dialog.

Once the database has been restored you can query the database. You can take a backup of a database from your test servers and restore it on your production server. In such a circumstance you might choose to skip the security information if the security defined on production servers is different from those on your test servers. In such a circumstance you would need to ensure you secure the database by defining the right security on production servers. In a circumstance where the backup was taken on your production server and you are restoring the database on an upgraded production machine we do expect users to restore the database with the security information.

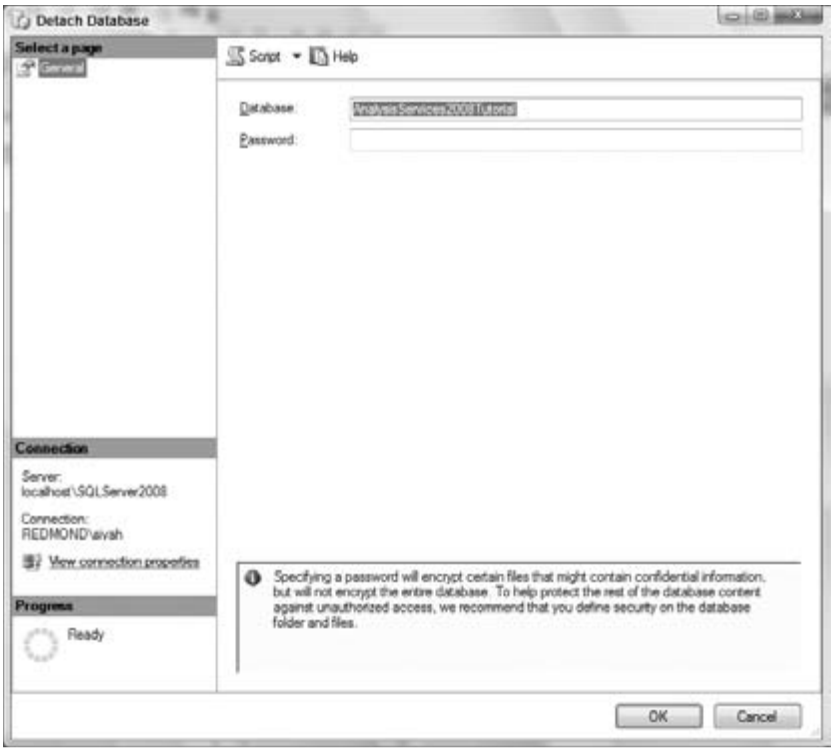
Detach and Attach

Analysis Services provides you the functionality to detach and attach a complete database from an Analysis Services instance. These detach and attach commands differ from backup and restore commands. The attach operation allows you to mark a specific database read - only, and the database ' s data files do not have to be stored in the default Data folder path of your Analysis Services instance. The read - only feature allows you to have a shared scalable architecture of Analysis Services for situations where you have a need to scale out the server to multiple users who are querying a specific Analysis Services database.

Follow these steps to detach the AnalysisServices2008Tutorial database:

1. In SSMS right - click the AnalysisServices2008Tutorial database and select Detach as shown in Figure 7 - 33 .

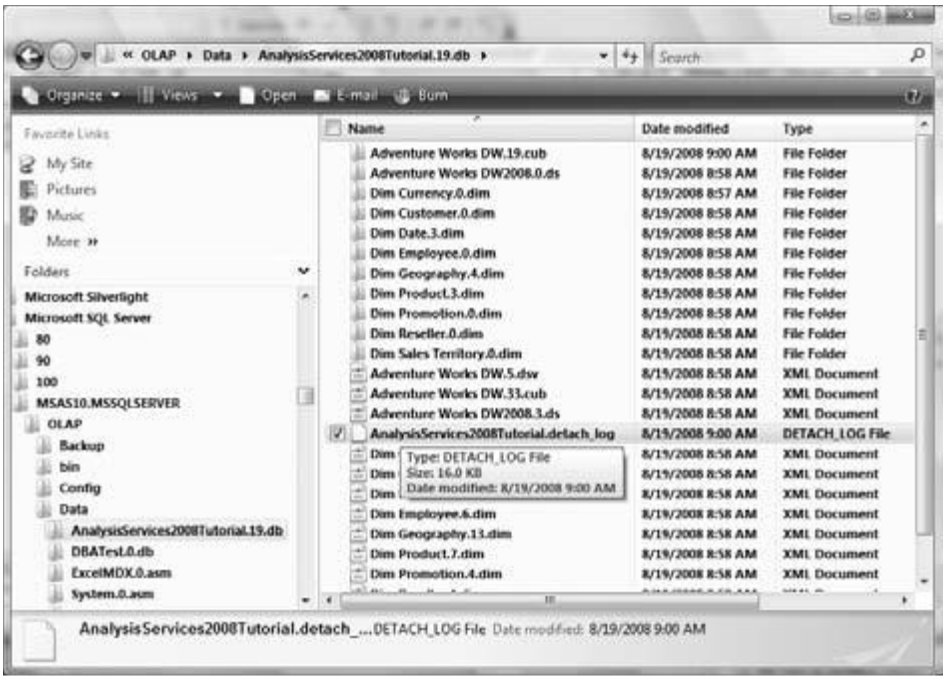
2. In the Detach Database dialog shown in Figure 7 - 34 , click OK.



SSMS sends the following XMLA command to the Analysis Services instance:

```
< Detach xmlns="http://schemas.microsoft.com/analysisservices/2003/engine" >
< Object >
< DatabaseID > AnalysisServices2008Tutorial < /DatabaseID >
< /Object >
< /Detach >
```

After receiving the detach command, Analysis Services first takes a write lock on the database to be detached. Taking a write lock means all existing DDL operations must complete before the detach command is started. The Analysis Services instance creates a detach log file that contains the version information, the key used for encrypting the database (if specified), and a few additional pieces of information about the database with the name AnalysisServices2008Tutorial.detach_log. This log file is created within the database folder as shown in Figure 7 - 35 . Analysis Services then commits and deletes the database. The entire database folder is now independent and can be copied and attached to another Analysis Services instance.

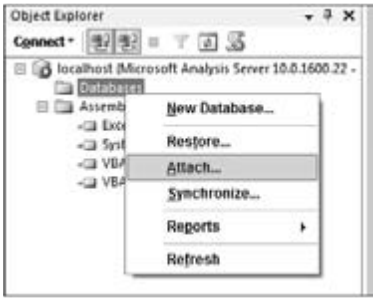


You can now attach the detached database to your Analysis Services instance. Follow these steps to attach the database in read - only mode:

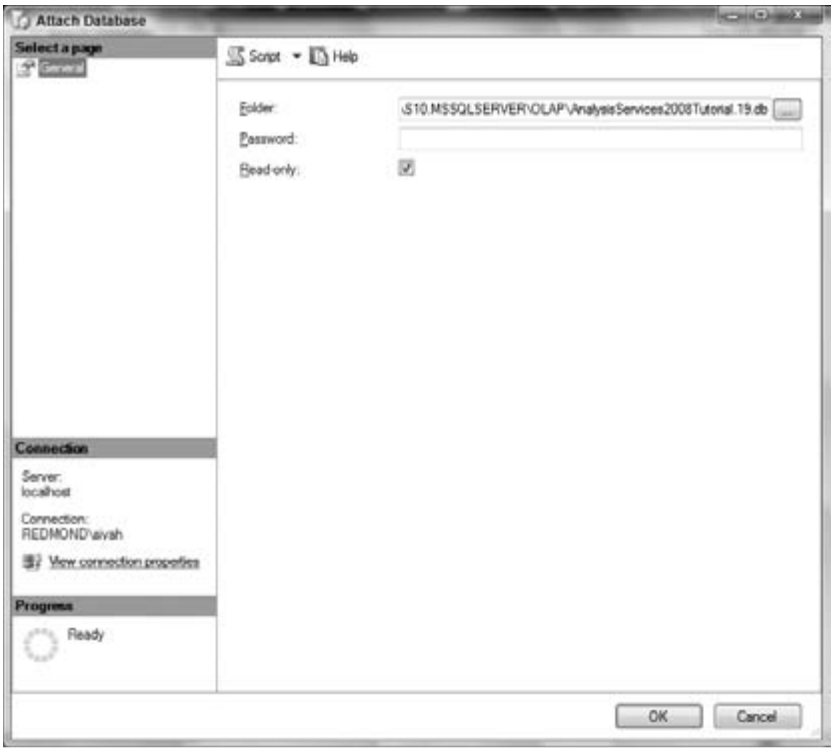
1. Move the AnalysisServices2008Tutorial database folder that was detached from its original location under %Program Files%\Microsoft SQL Server\MSAS10.SQLServer\OLAP\ Data to %Program Files%\Microsoft SQL Server\MSAS10.SQLServer\OLAP.

2. If prompted by the operating system to provide administrative privileges to move the folder, provide the permissions.

3. In the SSMS Object Explorer, right - click the Databases folder and select Attach as shown in Figure 7 - 36 .



4. In the Attach Database dialog, specify the full path of the AnalysisServices2008Tutorial database as shown in Figure 7 - 37 .



5. Enable the checkbox next to Read - only and click OK.

6. Refresh the Databases folder in the SSMS Object Explorer.

You will now see that the AnalysisServices2008Tutorial database has been attached to the Analysis Services instance. Because you attached the database as read - only, you will notice that this database has been marked in gray in the SSMS Object Explorer as shown in Figure 7 - 38 . You can also confirm that the database is read - only by right - clicking the database and selecting Properties. You will see the Read - Write Mode property set to ReadOnly in the Database Properties dialog. The read - only database feature in Analysis Services helps in having a shared scalable database architecture where you can have a single database folder on a Storage Area Network (SAN) attached to multiple Analysis Services instances. You learn how this is helpful in query performance in Chapter 15 .