Лекция 13

Тема « Administering Analysis Services »

Administering Analysis Services

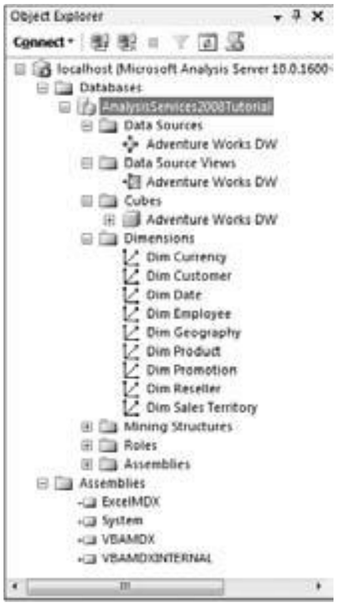
Administration is an important task on any server product. As an administrator of SQL Server Analysis Services (SSAS) you need to make sure Analysis Services is secure, reliable, and provides efficient access to the end users. You can administer AnalysisServices in two ways: through the SQL Server 2008 Tool set — SQL Server Management Studio (SSMS) and Business Intelligence Development Studio (BIDS) — or programmatically using an object model called AMO (Analysis Management Objects). You can accomplish tasks like processing objects, providing access to Analysis Services objects in databases, and synchronization of databases between Analysis Services instances using SSMS. You can use BIDS to connect to a specific OLAP database to perform design changes and accomplish follow - on tasks such as processing and providing access to users. SSMS and BIDS both use AMO behind the scenes to accomplish all management tasks.
The AMO object model itself is installed and registered into the GAC (Global Assembly Cache) when the product is installed. The AMO .NET assembly, by the way, is Microsoft.AnalysisSevices.dll.
In this chapter you learn about key administrative tasks and how to accomplish those tasks using SSMS and BIDS. In Chapter 13 you learn about administering SSAS programmatically using AMO.

Administration Using SQL Server 2008 Tools

Let ′ s just jump in and get our feet wet, shall we? In Chapter 2 you used SSMS to view the objects found in an Analysis Services 2008 database. We ′ ll start here on a similar footing:

1. Launch SSMS from All Programs Microsoft SQL Server 2008 SQL Server Management Studio.
2. Using Object Explorer, connect to the Analysis Services instance.
3. Open the Databases folder.

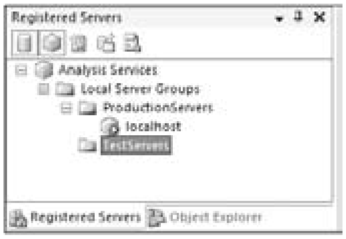
You will see a tree view of those databases you have saved on the server to date, as shown in Figure 7 - 1 . One of those databases should be titled AnalysisServices2008Tutorial — you should take a moment to review the tree nodes and what they contain because you will be learning the administrative tasks associated with those objects.



Managing Analysis Servers

SSMS, the integrated management environment for SQL Server 2008 products, provides you the flexibility of managing several Analysis Services instances. In this chapter we use the word “ server ” to denote an instance of Analysis Services, and “ servers ” to denote one or more. If you have a set of production servers that are being used by customers and a set of test servers that are being used for development and testing purposes, you typically want to manage them differently. The most logical thing is to group these servers. Using the Register Servers window of SQL Server Management Studio, you can group a set of Analysis Services servers to form a Server group as shown in Figure 7 - 2 . You can register Analysis Services servers and organize them into groups using the New Server Group and New Server Registration dialogs that can be launched by right - clicking the Local Server Groups folder under the Analysis Services folder in the Registered Servers window of SSMS.

Some of the common tasks of starting, stopping, restarting, and configuring Analysis Services servers can also be accomplished from the Registered Servers window. You can right - click the specific Analysis Services instance and choose the appropriate operation. In addition, you can switch to the Object Explorer window of the connected SSAS instance, or launch the MDX query editor or SQL Server Configuration Manager dialog from this window.

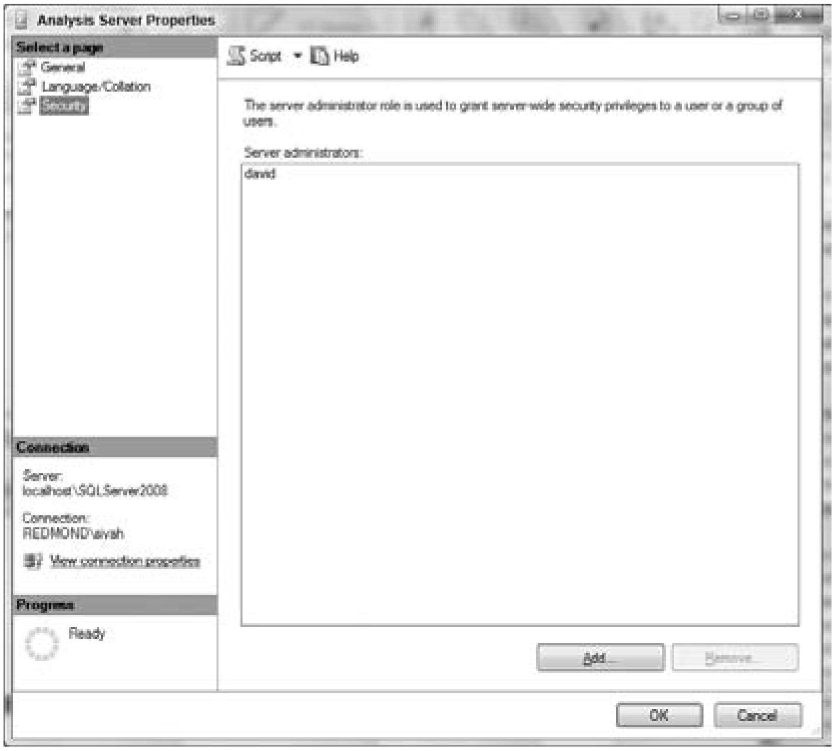


Once you are connected to an Analysis Services server in the Object Explorer window, you can accomplish various administrative tasks on that server, such as creating new databases, providing permissions, processing objects, and moving databases from test servers to production servers. First and foremost for the Analysis Server admin is providing access permissions to the users who will be administering the server. The following steps show how to add a user as an administrator of an Analysis Services server by making them part of the object called Server Role:

1. In the Object Explorer window right - click the Analysis Services instance and select Properties. You will now see the Analysis Services Properties dialog.

2. Click Security in the page as shown in Figure 7 - 3 .

3. Click the Add button to add a user to the Analysis Services administrators group. You can add domain users, local machine users, or groups as part of the administrator group for Analysis Services. If your user is a local user you can specify < machinename > \username or just the username to add the user to this server administrator group.



The server administrator role is used to grant server-wide security privileges to a user or a group of users.

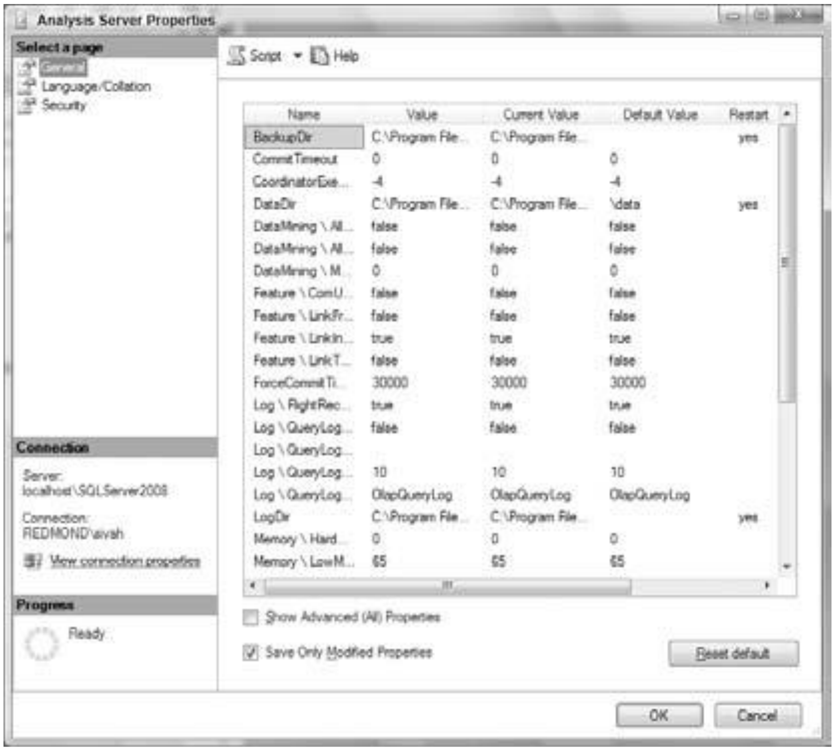
Server administrators:
david

Server:
localhost\SQLServer2008

Connection:
REDMOND\sivah

View connection properties

Another important management task is to set appropriate Analysis Server properties so that Analysis Services performs optimally. You can do this using the Analysis Server Properties dialog shown in Figure 7 - 4 . Analysis Services needs to be restarted for certain properties to take effect. This is indicated by a “ yes ” in the Restart column for those properties in the Analysis Services Properties dialog. Some of the most important properties involve control of parallelism for processing and querying and changing the read buffer size for faster query response time. Equally important are the maximum amount of memory used by the Analysis Services processes, the maximum number of connections to the server, and the ability to turn certain features on or off. You learn some of these properties in this chapter and others in Chapter 14 . The properties dialog has a checkbox that enables you to view and modify the advanced properties of the Analysis Services server. Adding users to the Server role or Database role and setting properties are considered part of securing your Analysis Services server. You learn more about managing security at the end of this chapter.



Server:
localhost\SQLServer2008

Connection:
REDMOND\sivah

Name	Value	Current Value	Default Value	Restart
BackupDir	C:\Program File...	C:\Program File...		yes
CommitTimeout	0	0	0	
CoordinatorExe...	-4	-4	-4	
DataDir	C:\Program File...	C:\Program File...	\data	yes
DataMining \ All...	false	false	false	
DataMining \ All...	false	false	false	
DataMining \ M...	0	0	0	
Feature \ ComU...	false	false	false	
Feature \ LinkFr...	false	false	false	
Feature \ LinkIn...	true	true	true	
Feature \ LinkT...	false	false	false	
ForceCommitTi...	30000	30000	30000	
Log \ FlightRec...	true	true	true	
Log \ QueryLog...	false	false	false	
Log \ QueryLog...				
Log \ QueryLog...	10	10	10	
Log \ QueryLog...	OlapQueryLog	OlapQueryLog	OlapQueryLog	
LogDir	C:\Program File...	C:\Program File...		yes
Memory \ Hard...	0	0	0	
Memory \ LowM...	65	65	65	

□ Show Advanced (All) Properties

☑ Save Only Modified Properties

Managing Analysis Services Objects

Several management tasks can be performed on Analysis Services objects. Some of the most important tasks are processing cubes and dimensions, providing access permissions to various objects within a database, managing the partitions of a cube based on usage, and adding assemblies to databases. Even though the SQL Server Management Studio provides a great interface to manage Analysis Services 2008 and abstracts all the internal details, it is beneficial to understand the underlying operations that take

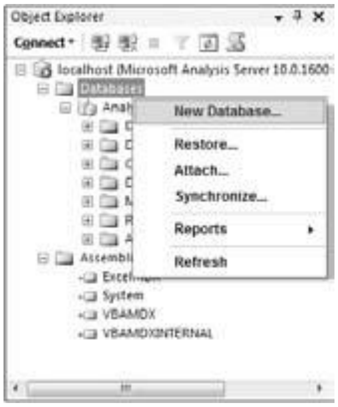
place when you perform the management operations. Knowledge of these server internals gives you an edge in better understanding its operation and helps you more effectively manage the server when unforeseen problems occur.

All communications to Analysis Services is through XML for Analysis (XMLA). The management tasks executed through SSMS use the management object model AMO (Analysis Management Objects), which in turn sends XMLA Execute commands to the Analysis Services instance. You will see some of the commands sent to the Analysis Services server when performing management tasks in this chapter.

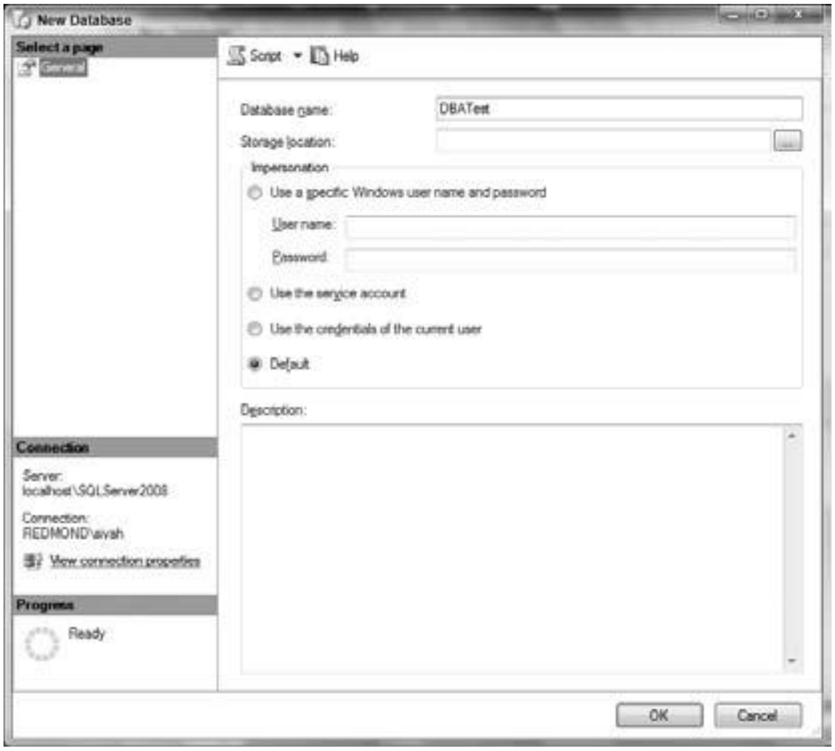
Database Creation

SQL Server Analysis Services 2008 allows a server administrator to create databases and assign database administrative permissions to a user. The following steps show how to do this:

1. In the SSMS Object Explorer, right - click the Databases folder and select New Database as shown in Figure 7 - 5 .



2. Enter a new database name called DBATest as shown in Figure 7 - 6 and click OK. SSMS sends an XMLA command to SSAS to create the new database called DBATest. SSMS then refreshes the Databases folder by retrieving the list of Databases from SSAS. You should see the DBATest database as shown in Figure 7 - 7 . If you are an administrator of SSAS, your account is a member of the Analysis Services server administrator role as seen in Figure 7 - 3 . If you want to provide a user with database administrator privileges and not Analysis Services server - wide privileges, you need to provide appropriate permissions at the database level. Follow the next steps to provide database administrator permissions for a user.



Database name: DBATest

Storage location:

Impersonation

○ Use a specific Windows user name and password

User name:

Password:

○ Use the service account

○ Use the credentials of the current user

● Default

Description:

Connection

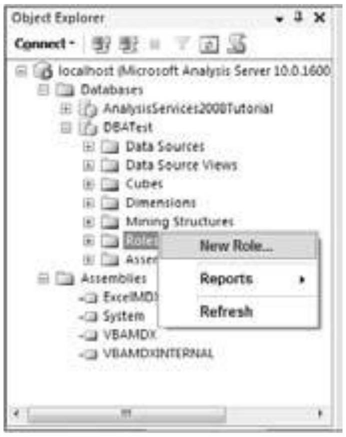
Server:
localhost\SQLServer2008

Connection:
REDMOND\sivah

View connection properties

Progress

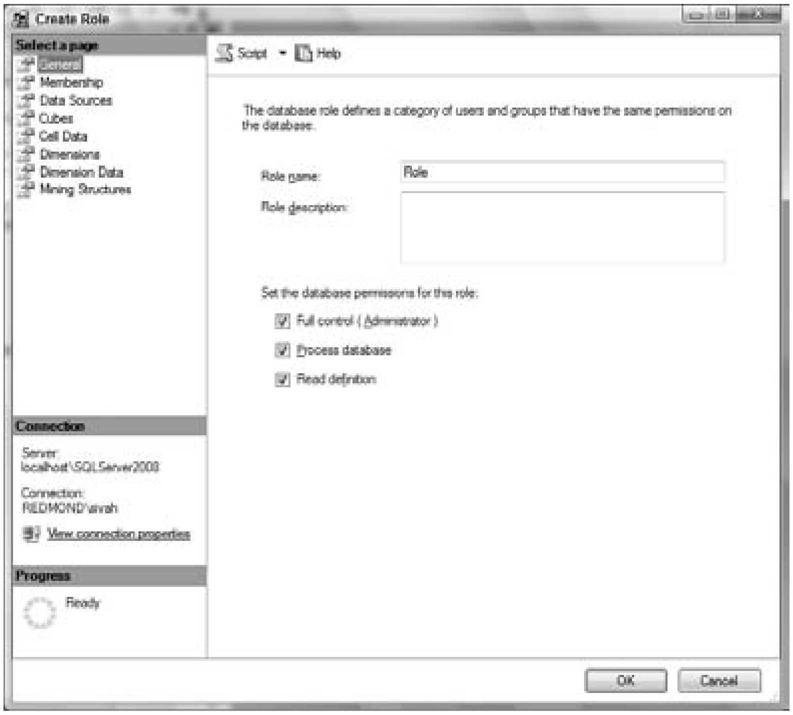
Ready



3. Expand the folder DBATest. You will see the various folders under DBATest.

4. Right - click the folder Roles and select " New Role " as shown in Figure 7 - 7 .

5. In the Create Role dialog, check the " Full control (Administrator) " checkbox (shown in Figure 7 - 8) to provide full database administrator privileges.



The database role defines a category of users and groups that have the same permissions on the database.

Role name: Role

Role description:

Set the database permissions for this role:

☑ Full control (Administrator)

☑ Process database

☑ Read definition

Connection

Server:
localhost\SQLServer2008

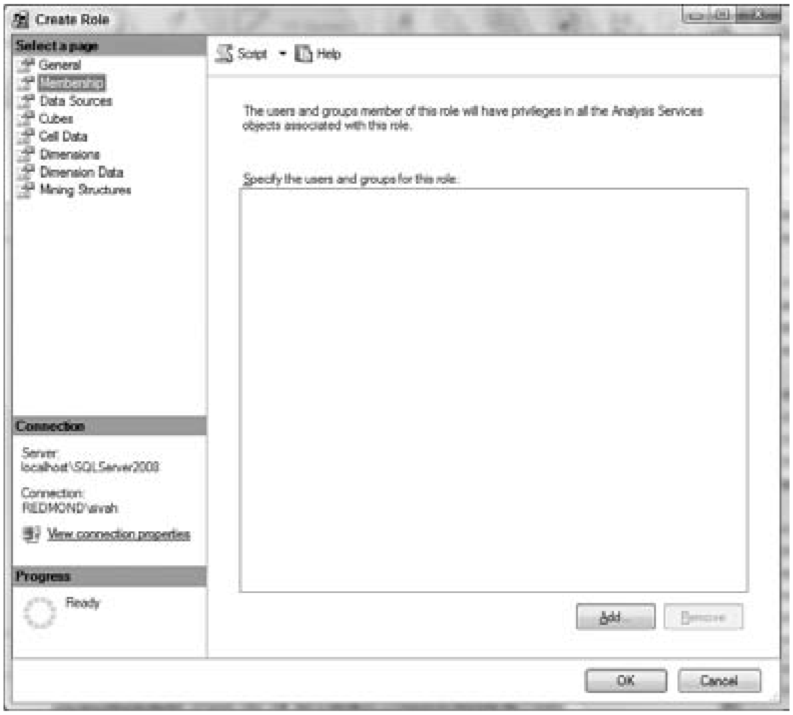
Connection:
REDMOND\sivah

View connection properties

Progress

Ready

6. Select the Membership page in the dialog as shown in Figure 7 - 9 .



The users and groups member of this role will have privileges in all the Analysis Services objects associated with this role.

Specify the users and groups for this role:

Server:
localhost\SQLServer2008

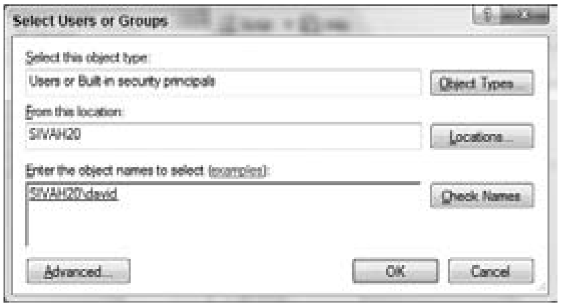
Connection:
REDMOND\sivah

View connection properties

Ready

7. Click the Add button on the Membership page to bring up the Select Users or Groups dialog.

8. Enter the user or users for whom you want to provide database permissions as shown in Figure 7 - 10 and click OK. You should now see the user you specified in the list of users who will have database permissions listed in the Create Role dialog.



Select this object type:
Users or Built-in security principals

From this location:
SIVAH20

Enter the object names to select (examples):
SIVAH20\david

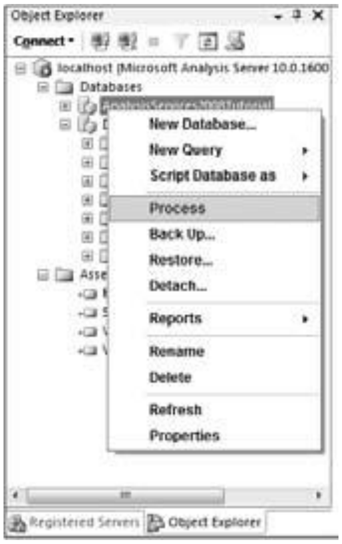
9. Click OK in the Create Role dialog.

You have successfully created a database called DBATest and provided full database permissions to a specific user. The user listed under the role Role will have full permissions to modify any of the objects that are part of the database DBATest including deleting the database. This user does not have permissions to perform any database operations outside of the DBATest database unless the same user is part of the Analysis Server administrator role. To create data sources, Data Source Views, cubes, dimensions, and mining models we recommend creating/modifying the objects in the DBATest using the online mode in Business Intelligence Development Studio (BIDS). You learn to work with SSAS databases using the online mode of BIDS later in this chapter.

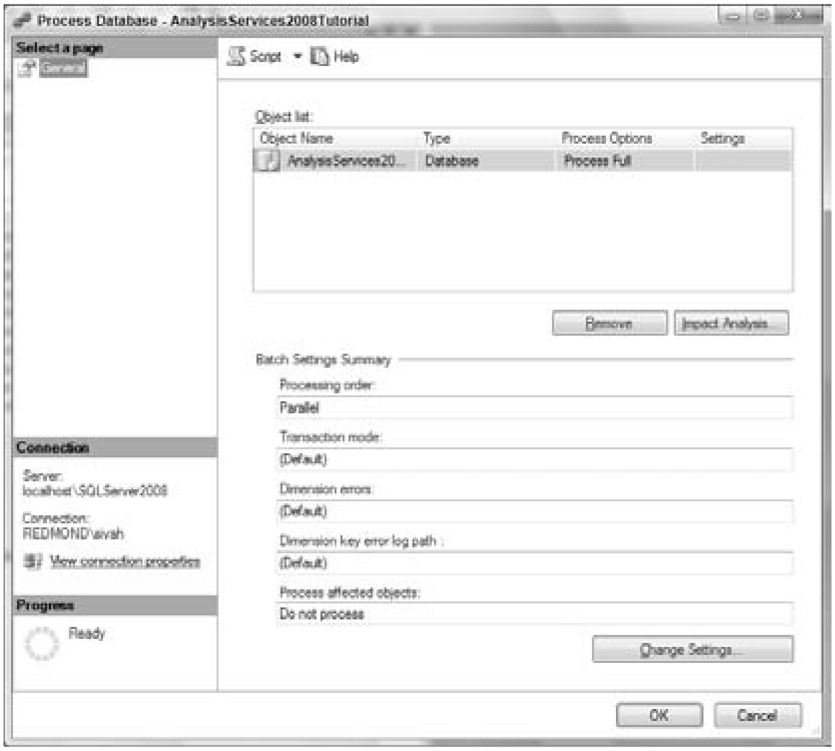
Processing Analysis Services Database Objects

One of the important jobs of an Analysis Services DBA (database administrator) is to process the objects (such as Cubes, Dimensions, and Mining Models) in an Analysis Services database. Analysis Services 2008 provides fine - grain control to the Analysis Services DBA to process the objects within an Analysis Services database using the Process dialog. You can launch the Process dialog by right - clicking the object folders such as Cubes, Dimensions, and Mining Structures — this works just as well on individual objects or groups of objects too. Based on the location from which the Process dialog is launched, the options for processing the object or group of objects will vary. In addition to this you can select an object and launch the Process dialog. To process the database AnalysisServices2008Tutorial, do the following:

1. Right - click the database AnalysisServices2008Tutorial and select Process as shown in Figure 7 - 11 .



You will see the Process dialog as shown in Figure 7 - 12 . This dialog shows the name of the object to be processed along with the type of object. Several processing options are available for each object. The default option for the database object is Process Full. As the name implies, the Process Full option allows you to process the selected object completely even if the object had been processed earlier. It will clear any data that was processed earlier.


Process Database - AnalysisServices2008Tutorial

Object list:

Object Name	Type	Process Options	Settings
AnalysisServices20...	Database	Process Full	

Batch Settings Summary

Processing order:
Parallel

Transaction mode:
(Default)

Dimension errors:
(Default)

Dimension key error log path :
(Default)

Process affected objects:
Do not process

Connection

Server:
localhost\SQLServer2008

Connection:
REDMOND\sivah

Progress

Ready

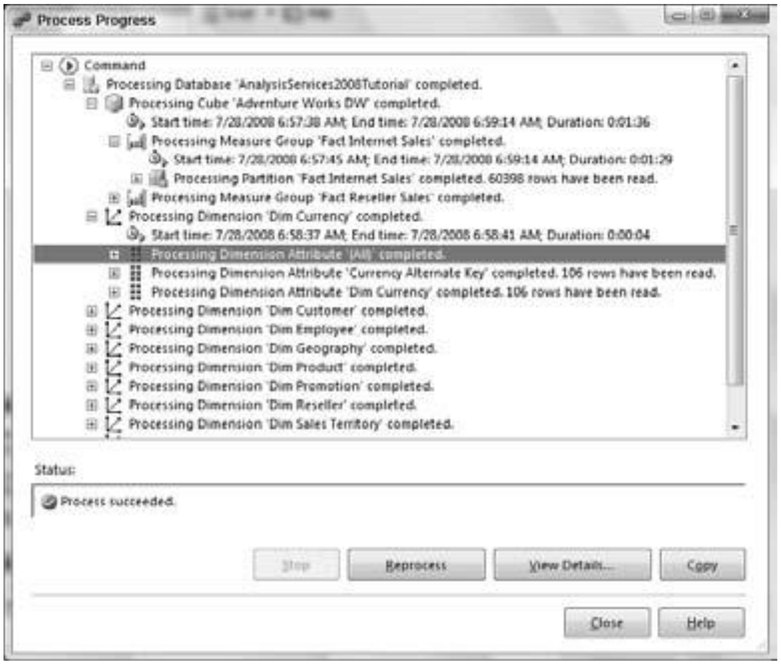
2. When you click the OK button the Process dialog sends an XMLA command to the Analysis Services instance to process the selected object. If you click on the Script button shown in Figure 7 - 12 and then select Script Action to New Query Window, you can see the Process XMLA command to be sent to the Analysis Services instance. You will see the following script command:

```
< Batch xmlns="http://schemas.microsoft.com/analysisservices/2003/engine" >
< Parallel >
< Process 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" >
< Object >
< DatabaseID > AnalysisServices2008Tutorial < /DatabaseID >
< /Object >
< Type > ProcessFull < /Type >
< WriteBackTableCreation > UseExisting < /WriteBackTableCreation >
< /Process >
< /Parallel >
< /Batch >
```

3. Click OK in this dialog to process the AnalysisServices2008Tutorial database. When you click OK the Process dialog uses AMO to send the Process command to the Analysis Services instance.

The Process XMLA script contains several commands that are interpreted by Analysis Services. Because the medium of communication to Analysis Services is an XMLA request, the script is embedded within SOAP Envelope tags. This script can be executed from the XMLA editor within SQL Server Management Studio. SSMS adds the appropriate SOAP envelope tags to send the script to Analysis Services. The commands in the script are Batch, Parallel, and Process. The Process command is part of a set of commands that manipulate the data in Analysis Services. These commands that change the data in Analysis Services databases are called the DML (data manipulation language). The Batch command allows multiple commands to be executed within a single statement. The Parallel command allows you to instruct the Analysis Services instance to execute all the commands within the command in parallel. The Process command is used to process an Analysis Services object and needs several properties such as DatabaseID, Process Type, and processing options (not shown in the above XMLA script) such as parallelism for processing objects, and actions to be taken during dimension key errors that can be changed using the Change Settings button in the Process dialog . You learn the processing options provided by the Process dialog in this chapter.

As mentioned earlier, when you click OK in the Process dialog, a Process command with appropriate options is sent to the Analysis Services instance. This command requests the server to process the database. When processing the objects within a database, the server needs to read data from the data source, which is done by issuing queries to it. You will now see the Process Progress dialog that shows details of each processing operation on the server. As you can see from Figure 7 - 13 , the operations on each object within the database that is being processed are reported along with the timing information and whether the operation succeeded or failed. You can also see the query sent to the relational data source to retrieve the data. The detailed information returned from Analysis Services is very helpful if you need to investigate any issues in processing including the performance of processing an object.



```
Command
  Processing Database 'AnalysisServices2008Tutorial' completed.
    Processing Cube 'Adventure Works DW' completed.
      Start time: 7/28/2008 6:57:38 AM; End time: 7/28/2008 6:59:14 AM; Duration: 0:01:36
      Processing Measure Group 'Fact Internet Sales' completed.
        Start time: 7/28/2008 6:57:45 AM; End time: 7/28/2008 6:59:14 AM; Duration: 0:01:29
        Processing Partition 'Fact Internet Sales' completed. 60398 rows have been read.
      Processing Measure Group 'Fact Reseller Sales' completed.
    Processing Dimension 'Dim Currency' completed.
      Start time: 7/28/2008 6:58:37 AM; End time: 7/28/2008 6:58:41 AM; Duration: 0:00:04
      Processing Dimension Attribute '(All)' completed.
      Processing Dimension Attribute 'Currency Alternate Key' completed. 106 rows have been read.
      Processing Dimension Attribute 'Dim Currency' completed. 106 rows have been read.
    Processing Dimension 'Dim Customer' completed.
    Processing Dimension 'Dim Employee' completed.
    Processing Dimension 'Dim Geography' completed.
    Processing Dimension 'Dim Product' completed.
    Processing Dimension 'Dim Promotion' completed.
    Processing Dimension 'Dim Reseller' completed.
    Processing Dimension 'Dim Sales Territory' completed.
```

Status:

Process succeeded.

Once all the objects have been processed you will see the results of the processing command. If all the objects were successfully processed, you will see Process succeeded in the status as shown in Figure 7 - 13 . If there were errors during processing, the status bar will show an appropriate message. The operations that resulted in an error are shown in red in the tree view of the Process Progress dialog. You can drill down into the details of the processing to understand the reasons for failure.

Several operations take place in the preceding processing command. All the objects within the database are processed in parallel based on the settings of the Analysis Services instance. If there are dependencies, the dependent objects are processed first. For example, the dimensions that are part of a cube need to be processed before the cube can be processed. Analysis Services processes all the objects of the database under a single transaction. What this means is that if one of the objects failed during processing, the remaining objects will not be processed and the effects of any previous operations will be rolled back. For example, if all the dimensions of a cube were successfully processed and if there were errors while processing the cube, the processing of the dimension objects will be rolled back. Once all the objects have been successfully processed, the server commits the transaction, which means that the objects are marked as processed and are available for querying.

Assume an Analysis Services object has been processed and is being queried by users. At the time users are querying the object, you can initiate processing on the same object. Because a version of the object is currently being queried, Analysis Services stores the uncommitted processed object in a temporary file. At the time of commit, the server first ensures that the user is not using the objects, removes the previous version of the processed objects, and then marks the temporary files as primary. You see this in detail in the following section.