

## CHAPTER 4

# Inside Management and Administration Tools

The Microsoft SQL Server Team had two main goals for the design of the new SQL Server 2005 management tools. The first goal was to provide a consistent management experience for all components of SQL Server. The second goal was to provide an innovative tool to make database professionals more productive at querying, managing, and administrating SQL Server. To achieve these goals, the team had to choose an environment that was easily extensible, user-friendly, and familiar to the database professionals. While we were studying different hosting environments, or as we called them “shells,” we soon concluded that the Microsoft Management Console (MMC) shell was not suitable to host the new tools. The MMC model was old, monolithic, and it did not provide the extensibility needed. For example, the MMC model stipulated that the management dialogs had to be modal with a fixed size optimized for a desktop size of  $640 \times 480$  pixels. However, as the high-resolution computer monitors became more popular and the average size of the desktop increased to  $1024 \times 768$ , the legacy Enterprise Manager users hated this restriction. They consistently asked us why they could not resize the management dialogs to take advantage of the full size of their screens. However, deviating from the MMC model was not that easy. A new shell would definitely solve some of the issues, but it would also break a long-term trend. For two generations of SQL Server, database professionals had used the Enterprise Manager or Analysis Manager, both of which were MMC snap-ins. The innovative new shell had to eliminate some of the legacy restrictions but also had to be familiar to users.

One of the environments that looked very attractive was the Integrated Development Environment (IDE) upon which Visual Studio was built. The IDE had provided extensibility and a proven, user-friendly environment for developers for several years. It also offered several engineering advantages, such as a ready-to-use editor, a debugger, and a

## 130 Chapter 4 Inside Management and Administration Tools

---

powerful help model, and it could host some of the components that were built by the Visual Studio team. After studying several prototypes, the team decided to use the IDE for the new shell. The tool was called SQL Server Workbench and it was distributed to a number of early adopter customers and SQL Server Most Valued Professionals (MVPs). By the Beta 1 release, SQL Server Workbench created a lot of controversy in the community. The feedback was very bipolar; customers either loved or hated it. While the developers found the environment extremely friendly, the DBAs found the user interface confusing. Some DBAs thought the shift to a development-like environment unnecessary, and it required a steep learning curve. The engineering team faced a tough task addressing these issues. The decision was to cut down some of the features and use the resources to address these important usability issues. The team carefully collected all the feedback and redesigned the environment to bridge the gap between the legacy tools and the new innovative tool. We also made the environment simpler and friendlier for the DBAs. By the Beta 2 release, the tool was called SQL Server Management Studio, and it received positive feedback from both developers and DBAs. The results were successful, but it came at the price of cutting some features such as the T-SQL debugger and Object Search functionality.

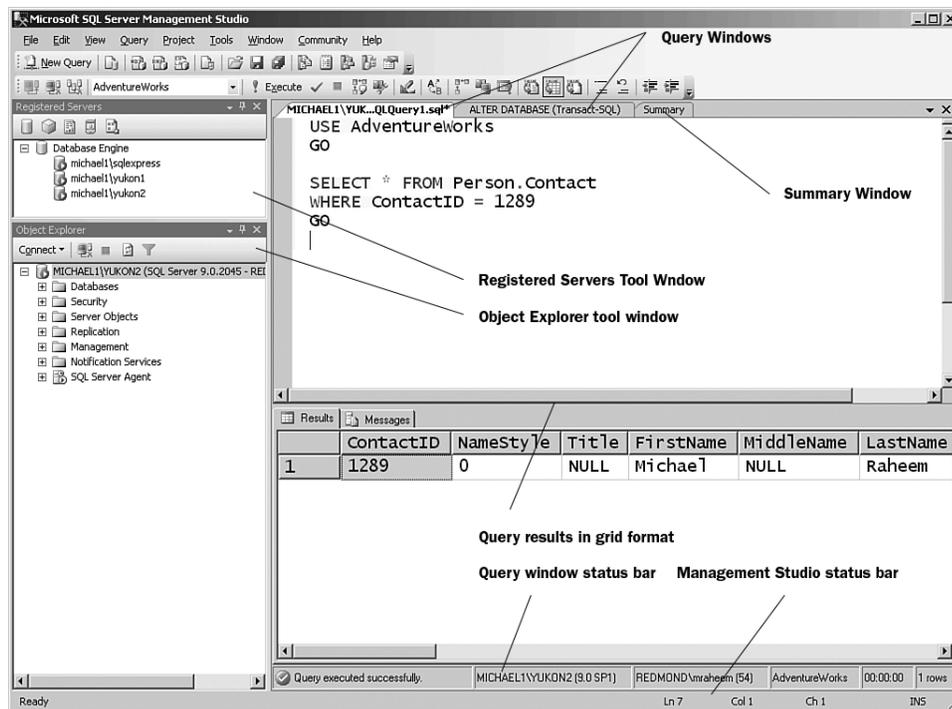
With almost six months since the release of SQL Server 2005 at the time of this writing, I (Michael) can see that the team made the right decision to achieve its goals. SQL Server Management Studio provides the ultimate integrated management and authoring experience for all components of SQL Server and it is a productive and user-friendly environment. You may not find all the functionality of Enterprise Manager or Query Analyzer in the new tool, but it certainly increases your productivity. As for the missing features, you can find them in other tools—as with the T-SQL debugger now found in Visual Studio—or look for them in the future releases of SQL Server. Remember, the SQL Server team is always looking forward to your feedback: <http://connect.microsoft.com>.

### **SQL Server Management Studio**

---

Management Studio is a comprehensive integrated environment to manage, administer, and author queries for all components of SQL Server, namely Database Engine (including Replication, SQL Server Agent, and Full-Text Search), Analysis Services, Reporting Services,

Integration Services, Notification Services, and SQL Server Mobile. Management Studio is a collection of rich tools in the form of dialogs, wizards, tool windows, designers, and editors (see Figure 4-1).



**Figure 4-1** SQL Server Management Studio and some of its main components.

The Management Studio environment or “shell” leverages the same technologies as the Visual Studio Integrated Development Environment (IDE). However, Management Studio and Visual Studio are two completely different products. Management Studio is specifically designed for management and authoring of SQL Server products, whereas Visual Studio is an integrated development environment for all types of development such as Windows, Office, web, and mobile applications.

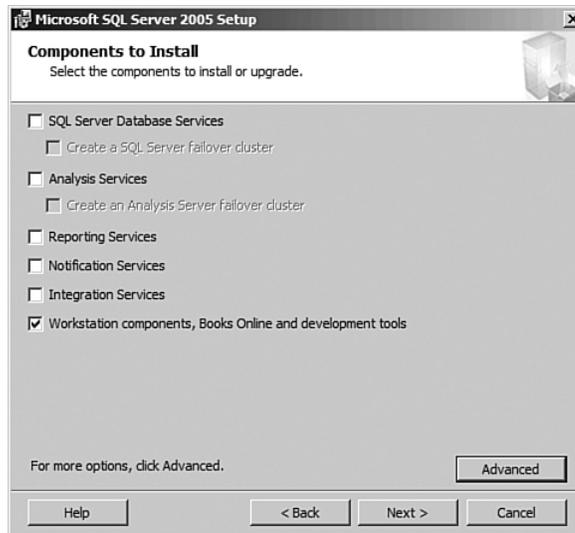
One of the main productivity gains in Management Studio is the integration of authoring and management in a single environment. For those of you who are familiar with SQL Server 2000/7.0 tools, Management Studio combines the functionality of Enterprise Manager, Query Analyzer, MDX Sample Application, and Analysis Manager in a single

## 132 Chapter 4 Inside Management and Administration Tools

environment. In addition to SQL Server 2005, SQL Server Management Studio can manage SQL Server 2000 Database Engine.

### Installing and Launching SQL Server Management Studio

SQL Server Management Studio can connect locally or remotely to a server. To install Management Studio, run the SQL Server Setup program and proceed with the setup screens until you see the **Components to Install** screen. On this screen, select **Workstation components, Books Online and development tools** (see Figure 4-2), and continue with the rest of the setup screens.



**Figure 4-2** Management Studio can be installed as part of the workstation components from the SQL Server setup program.

#### NOTE

The option, Workstation components, Books Online and development tools in the setup program installs more than just Management Studio. If you want to install only the management tools, click on Advanced and exclude all other components but Management Tools in the subsequent screen.

To launch Management Studio from Windows Start menu, point to **All Programs/Microsoft SQL Server 2005**, and select **SQL Server Management Studio**.

## Management Studio Components

Management Studio hosts a number of components for the authoring, administration, and management of SQL Server products. The following sections present a description of the main components in Management Studio along with their functionality.

### Connection Dialog

Connection dialog provides the functionality to specify the necessary parameters to connect to a server (see Figure 4-3).



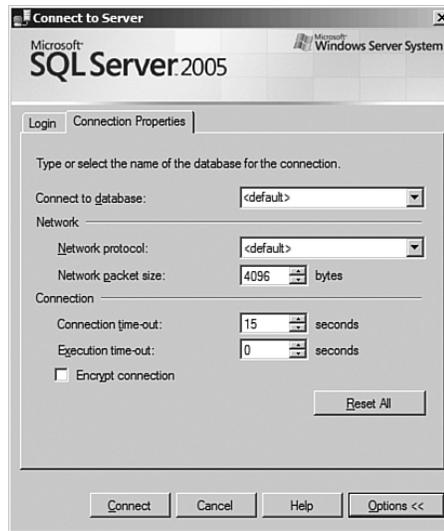
**Figure 4-3** Connection dialog in SQL Server Management Studio.

Using the fields **Server type** and **Server name**, you can connect Management Studio to any of the SQL Server products on a given server. These products could be Database Engine, Analysis Services, Reporting Services, Integration Services, or SQL Server Mobile. As described in Chapter 2, you can install named instances of Database Engine, Analysis Services, and Reporting Services. To connect to a default instance of any server type, you can enter the name of the server in **Server name**. To connect to a named instance, enter the name of the server followed by the \ character and then the name of the instance. For example, in Figure 4-3 the user has entered MICHAEL\YUKON1 in the **Server name** field to connect to a database engine instance named YUKON1 on server MICHAEL1.

**134 Chapter 4 Inside Management and Administration Tools****NOTE**

You can substitute the full server name with (local) or . (dot character) to connect to a local server. For example, in Figure 4-3 if Management Studio were running on the same computer as MICHAEL1, you could enter (local)\YUKON1 or .\YUKON1 in the Server name field of the connection dialog.

On the connection dialog, you can optionally click on the **Options** button to access the advanced connection options (see Figure 4-4). You can find a description of these options in SQL Server Books Online. For most scenarios, you do not need to change any default values.

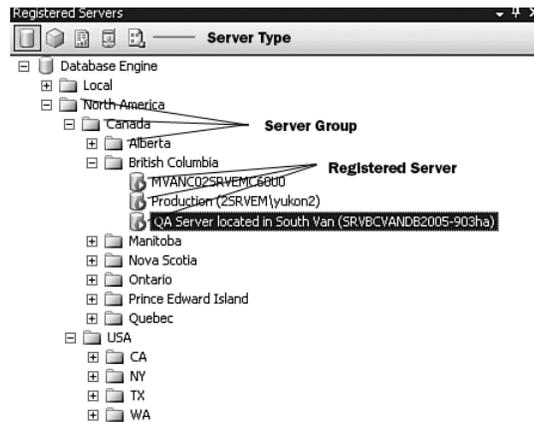


**Figure 4-4** Advanced connection options for SQL Server Database Engine.

Upon clicking on **Connect**, the Connection dialog tries to establish a connection with the specified server. While the connection is being established, the narrow orange band on top of the dialog moves from left to right and you can click on **Cancel** to cancel the connection. During the connection cancellation, the narrow orange band changes its movement to the opposite direction, moving from right to left.

## Registered Servers

Registered Servers (see Figure 4-5) is a Management Studio tool window that provides the functionality to keep links to your frequently used servers. You can use these links to quickly view the status of the server, connect to the server to manage its objects, or to execute a query. Each user has a separate list of Registered Servers stored in a file on the local computer (the same computer that is running Management Studio).



**Figure 4-5** Registered Servers provides a list of the frequently used servers.

You can organize the Registered Servers in server groups (similar to the files and folders) to match the logical or physical grouping of the servers in your organization. For example, in Figure 4-5 the registered servers are grouped by physical location of the servers—in other words, by continent, country, region, state, and so on. You can create registered servers for all server types. The toolbar on top of the Registered Servers window allows quick switching between different server types: Database Engine, Analysis Server, Reporting Services, SQL Server Mobile, and Integration Services. Alternatively, you can use **View, Registered Server Types** from the main menu in SQL Server Management Studio to switch between server types.

### TIP

You can create Registered Servers for SQL Server 2000 and SQL Server 2005 instances. However, there are no iconic or graphical representational differences between a SQL Server 2000 and SQL Server 2005 registered

## 136 Chapter 4 Inside Management and Administration Tools

server. If you are planning to create registered servers for both your SQL Server 2000 and SQL Server 2005 servers, it is a good idea to create your SQL Server 2000 registered servers in a separate server group named something like “SQL2000.” This will help you to distinguish the SQL Server 2000 registered servers quickly.

By default, the Registered Server window appears on the top-left corner of Management Studio. You can easily move, hide, or autohide the window to allocate more space to other components. To quickly view the window, choose **Ctrl+Alt+G** or **View, Registered Servers** from the main menu.

The Registered Servers window enables you to view and change the state of a server from the registered server. To view the status of a server, look at the icon to the left of the registered server and match it to Table 4-1.

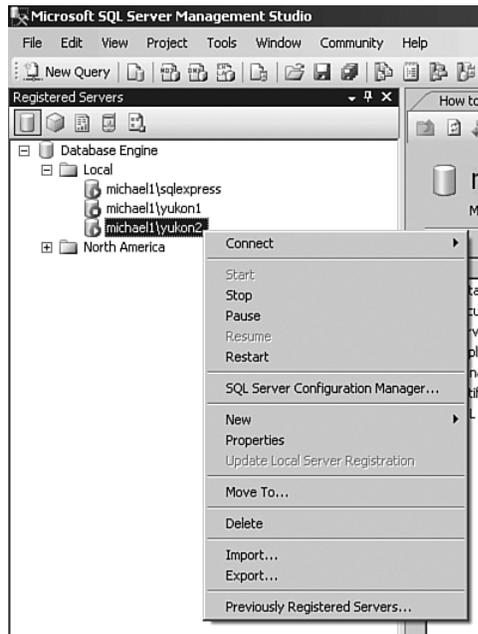
**Table 4-1** Status of Servers in the Registered Servers Window

Icon	Status	Description
	Started	Service is up and running.
	Stopped	Service is stopped.
	Paused	Service is paused.
	Unknown	The state of the service is unknown. Registered Server cannot access the service possibly because of network or WMI security issues—see “Troubleshooting Registered Servers to Show Server Status” in this chapter.

The context menu on a registered server (see Figure 4-6) provides the capability to change the state of a service, along with a variety of management options that are covered later in this chapter.

### **Object Explorer**

Object Explorer (see Figure 4-7) is one of the main components of SQL Server Management Studio. It provides a hierarchical tree view of the objects on a server. The tree displays nodes in folders representing the server’s logical structure. Object Explorer is the manifest for performing management operations on the server objects for Database Engine



**Figure 4-6** The Registered Servers menu provides a variety of management options.

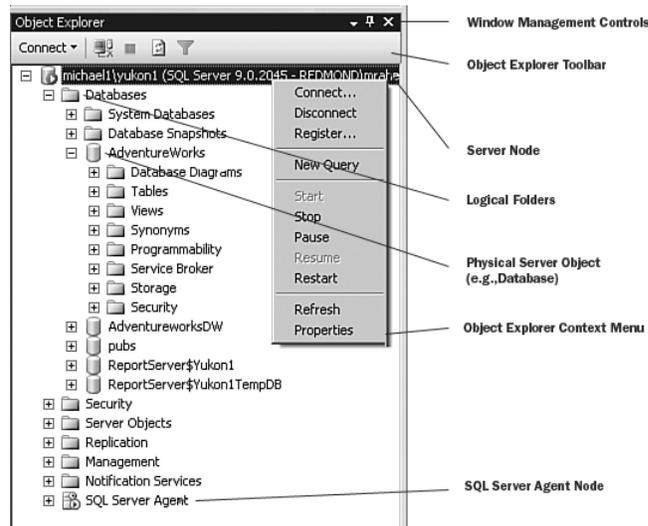
(including SQL Server Agent, Replication, and Full-Text Search), Analysis Services and Data Mining, Reporting Services, Integration Services, and Notification Services. Each folder or a node in Object Explorer has a context menu with a variety of options to perform an action on the server. For those of you who are familiar with SQL Server 2000 tools, Object Explorer is the replacement for the left tree in Enterprise Manager and Analysis Services, and Object Browser in Query Analyzer.

By default, the Object Explorer window appears on the left side of Management Studio, docked under the Registered Server window. You can move, hide, or autohide the window to allocate more space to other components. There are several ways to quickly show the Object Explorer window. You can press **F8** on the keyboard, select **View, Object Explorer** from the main menu, or click on the Object Explorer button on the main toolbar in Management Studio.

**NOTE**

Closing the Object Explorer window (clicking on the [x]) does not disconnect it from the server; it only hides it.

## 138 Chapter 4 Inside Management and Administration Tools



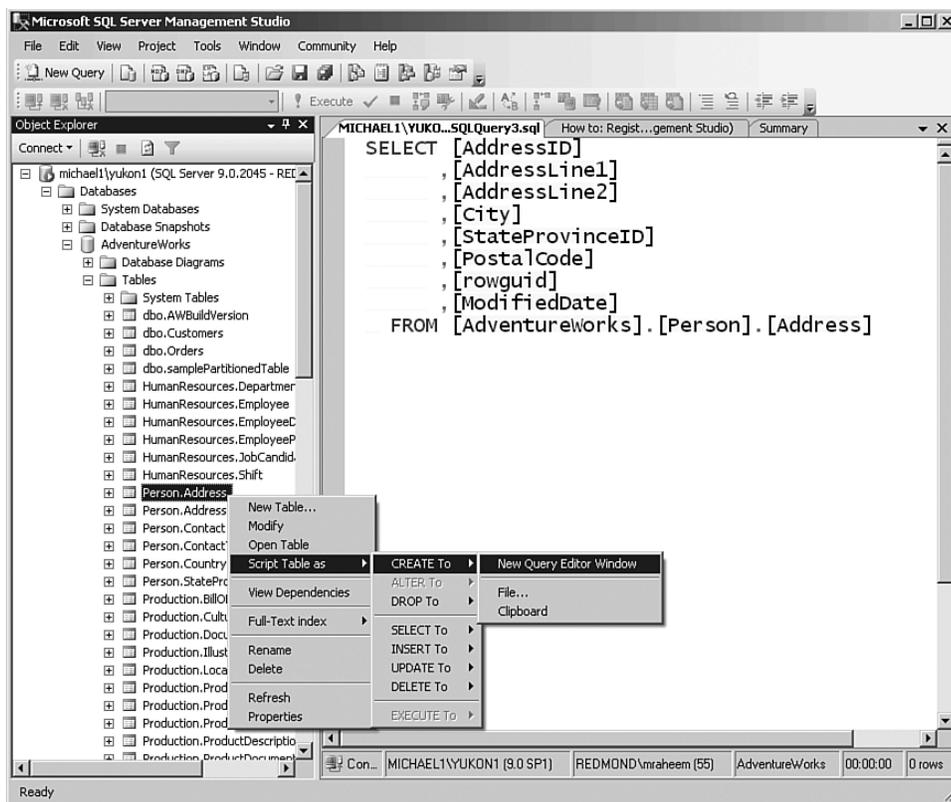
**Figure 4-7** Object Explorer and its various sub-components.

To connect Object Explorer to a server, you can right-click on a registered server and choose the option **Connect, Object Explorer**. You can also click on **Connect** in the toolbar of Object Explorer and select one of the server types. Alternatively, you can choose **File, Connect Object Explorer** from the Management Studio main menu. All these connection options, with the exception of the registered server, pop up the connection dialog, where you can enter the server name and other connection options. After you are connected, you can expand and collapse the nodes to navigate through the objects and use the context-menu to perform operations on the objects. To disconnect, click on **Disconnect** in the toolbar or right-click on the server node and choose the option **Disconnect**. The context menu of the server node has an option to register the server. This is particularly useful for situations in which you have connected to a server and you want to quickly add it to your registered servers for subsequent connections.

You can use Object Explorer to connect to multiple servers in the same window. The servers could be of the same type or different types, installed locally or remotely. This is a great productivity feature because you can manage all servers in your organization without launching a separate management tool or opening multiple instances of the same tool for each of the servers. In addition, Object Explorer supports connections to the SQL Server 2000 database engine. This is a useful option

during upgrade and migration processes, where you can use Object Explorer to manage both SQL Server 2000 and SQL Server 2005 environments and remove the old tools altogether.

In addition to managing objects, Object Explorer can assist you in programming and writing queries. As shown in Figure 4-8, you can right-click on an object (such as a table) and choose the option to create a DML (INSERT, UPDATE, DELETE), or a DDL (CREATE, ALTER, DROP) statement. You can optionally send the generated script to a new query window, a file, or the clipboard to subsequently paste into other editors. In addition, you can drag an object from Object Explorer to a query window rather than typing it. This is extremely useful in a situation in which you have objects with long names or names that are too difficult to spell.



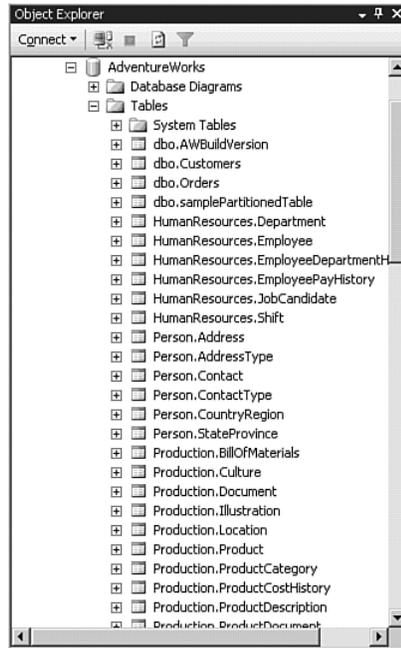
**Figure 4-8** Object Explorer provides the functionality to create DML and DLL statements for an object.

**140 Chapter 4 Inside Management and Administration Tools****NOTE**

You cannot select multiple objects in the Object Explorer window; use the Summary Window instead.

Object Explorer has several productivity features such as filtering. If you plan to work with a large number of objects, consider creating a filter. Filters enable you to view and work with only a subset of objects at a time. The filtering is applied just like a *WHERE* clause; it restricts the object list to only those objects that meet the filtering criteria. Let's go through a filtering example and see how it works:

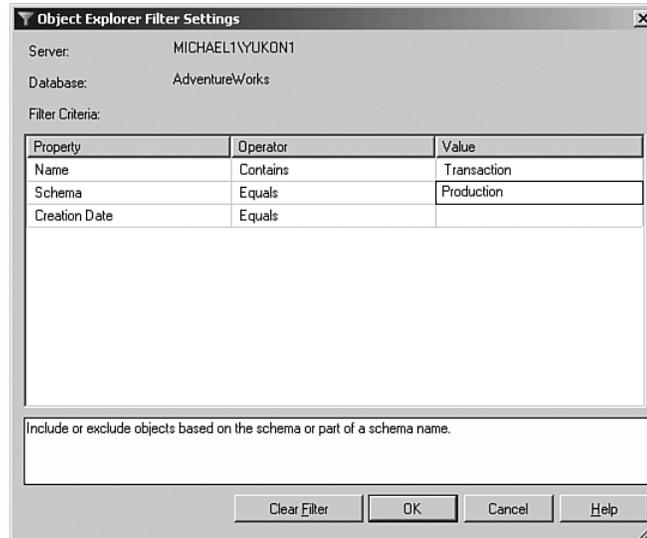
Figure 4-9 shows a snapshot of tables in the AdventureWorks database prior to filtering.



**Figure 4-9** Tables in AdventureWorks database prior to filtering.

Imagine that the objective is to create a filter to display only tables whose names contain the word “Transaction” and that belong to the “Production” schema. To create the filter, right-click on the **Tables** folder in Object Explorer and select **Filter, Filter Settings** in the context menu.

In the Object Explorer Filter Settings dialog, enter the filtering criteria to match the objective. For the **Name** property, pick the operator **Contains** and enter **Transaction** for its value. For **Schema**, pick the operator **Equals** and enter **Production** for the value. The finished filter settings are shown in Figure 4-10.



**Figure 4-10** Object Explorer filter settings for AdventureWorks tables.

After you click **OK**, the Object Explorer displays only two tables that matched the filtering criteria. Notice the word “filtered” is now appended to **Tables**, in parentheses, indicating that filtering is in effect (see Figure 4-11). To remove the filtering, you can right-click and select the option **Filter, Remove Filter**.

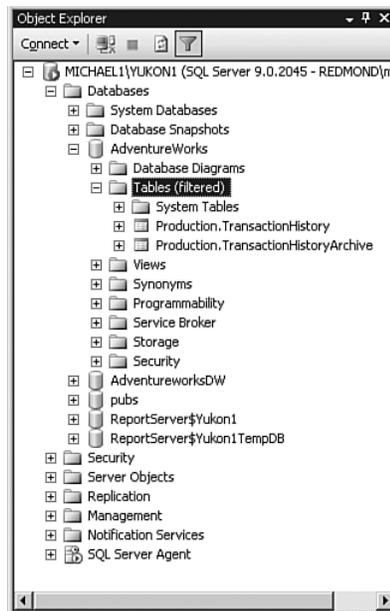
#### **NOTE**

The Object Explorer filter settings are removed and not persisted after you disconnect Object Explorer from the server.

Under the hood, Object Explorer uses a component called Enumerator to query the meta-data of objects on a server and displays them in a tree control. As the user clicks on each node of the server, Object Explorer sends a query that looks very much like an XML Path Language (XPath)

## 142 Chapter 4 Inside Management and Administration Tools

query to the enumerator. Enumerator interprets the XPath-like query into a metadata query, which is understood by the particular server type. The enumerator fetches the query result from the server and passes it to Object Explorer in XML format. Object Explorer transforms the XML into the graphic format and appends it to the appropriate node in the tree.

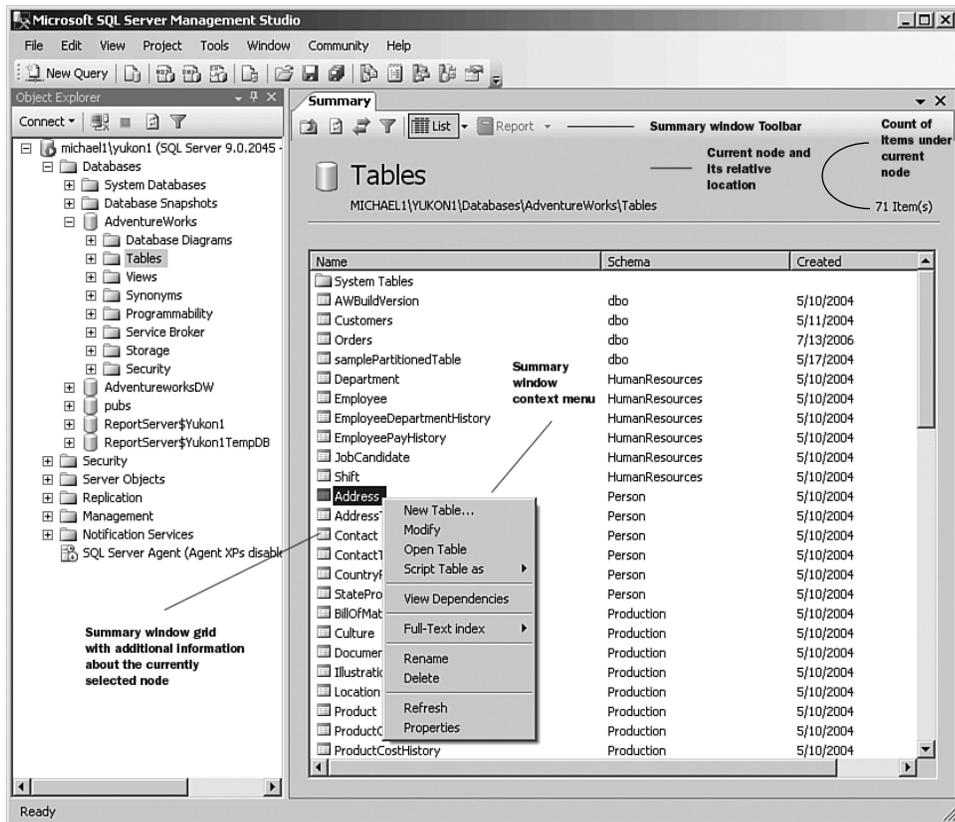


**Figure 4-11** Tables in AdventureWorks database after filtering.

### **Summary Window**

The Summary window (see Figure 4-12) works in conjunction with Object Explorer to display additional information about the selected object in Object Explorer. In addition, the Summary window provides an alternative method to Object Explorer for navigating through the objects in a server. You can find the title and relative location of the current object in the area just under the toolbar. The grid displays the down-level objects relative to the current object. You can double-click on the objects in the grid to further navigate their down-level objects. Double-clicking an object with no down-level objects opens a dialog where you can view/change properties of the object. To view the up-level node, click on the **Up** button in the Summary Window toolbar. Navigating through

Object Explorer automatically changes the current node in the Summary window, but not vice versa. To change the current node in Object Explorer to the same node in the Summary window, you need to click on the **Synchronize** button on in the Summary window toolbar. Just as you can with Object Explorer, you can use filtering by clicking on the **Filter** button in the Summary window toolbar.



**Figure 4-12** The Summary window provides an alternative method for viewing objects on the server with additional metadata.

The Summary window provides two views, **List** and **Details**, that can be selected from the **List** button on the toolbar. In Details mode, the Summary window grid displays additional metadata information for the down-level objects. You can sort the metadata information in the grid in

## 144 Chapter 4 Inside Management and Administration Tools

---

ascending or descending order by clicking on the column headers. This metadata information is useful for the times when you want to find the creation date of an object quickly or you need to find objects with the same name in different schemas.

---

### TIP

The Summary window performs better than Object Explorer for populations of nodes with many objects. The difference becomes more visible when populating nodes with 1,000 or more objects. It is recommended that you use the Summary window for navigation if the population of Object Explorer becomes a bottleneck.

---

### *Management Dialogs*

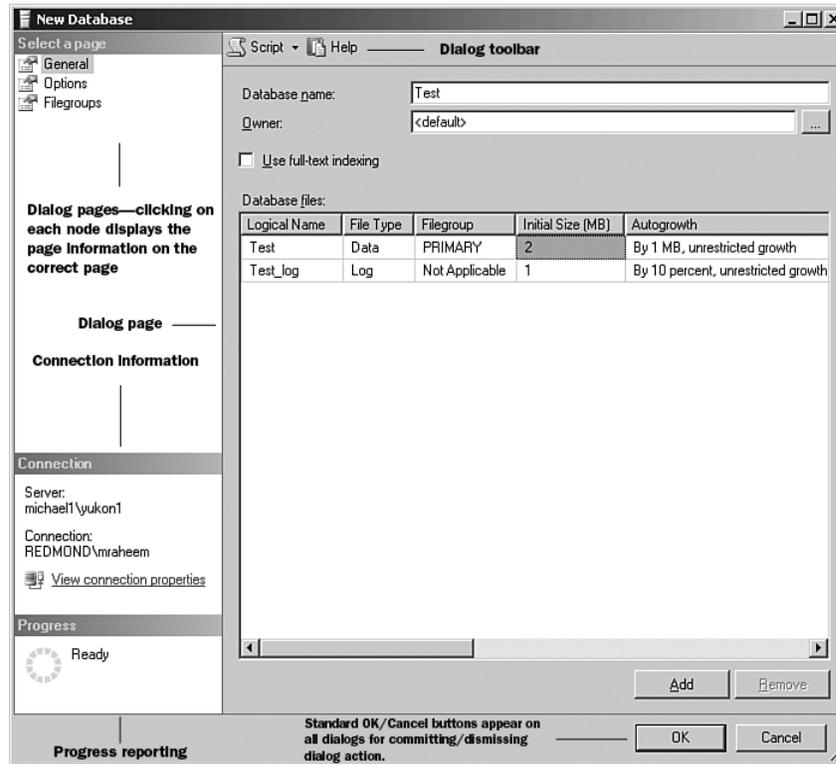
Management Dialogs are components of SQL Server Management Studio that provide the functionality to manage objects or perform actions on a server through a graphical user interface. There are over 200 management dialogs in Management Studio, which support all SQL Server 2005 products. Some of these dialogs are very simple and some are complex. However, they all follow a consistent layout and flow to ensure a low learning curve for the user. Figure 4-13 displays a typical management dialog and its various components.

You launch most of the management dialogs by selecting an entry on the Object Explorer context menu. The Object Explorer context menu is carefully designed to provide a link to dialogs and actions that are relevant to the currently selected node. For example, from the **Databases** folder in Object Explorer you can launch the following management dialogs that are relevant to databases:

- **New Database**—Allows creation of a new database.
- **Attach**—Creates a database from files that were previously used to store a database.
- **Restore Database**—Restores a database from a full or transaction log backup.
- **Restore File and Filegroups**—Restores specific files or file groups of a database.

The concept of management dialogs is not new in SQL Server 2005, and they have been around since SQL Server 6.0. However, a few innovations in SQL Server 2005 management dialogs could significantly

improve your productivity. These innovations provide a new generation of dialogs that are resizable, non-modal, scriptable, and schedulable.



**Figure 4-13** A typical management dialog in SQL Server Management Studio.

### Resizability

One of the strong bits of feedback from SQL Server 2000 customers was related to the shortcomings of management dialogs in Enterprise Manager. The dialogs were too small and not resizable. New in SQL Server 2005, almost all management dialogs are resizable. You can discover the resizability of a dialog from the size-grip  displayed on the bottom-right corner of the dialog.

### Non-Modality

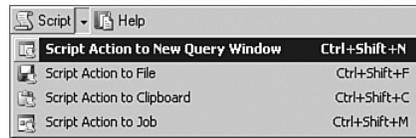
Another problem with SQL Server 2000 Enterprise Manager dialogs was their modality: Only one dialog could be opened at any given time. This was extremely frustrating in situations where the user was in the middle of

## 146 Chapter 4 Inside Management and Administration Tools

creating an object and wanted to look up properties of another object. To do this, the user had to cancel the first dialog, which meant losing all the data already entered. Next, the user had to open up the second dialog, look up the information, close the second dialog, and open the first dialog again and re-enter the information. This was a big hindrance to productivity. New in SQL Server 2005, multiple instances of management dialogs can be opened without any stipulation to close the previous ones.

### Scriptability

Most management dialogs provide several scripting options that can be accessed from the management dialog toolbar (see Figure 4-14).



**Figure 4-14** The management dialog toolbar provides various scripting options

The scripting options provide a means of generating a script for the action that the dialog is about to perform on the server. The language of the generated script depends on the type of server to which the management dialog is connected. Table 4-2 describes the script type for each of the server types.

**Table 4-2** Server Types and Script Languages Generated by the Management Dialogs

Server Type	Script Language and Execution Environment
Database Engine (including SQL Server Agent and Full-Text Search)	Transact-SQL; the generated T-SQL script can be executed in Management Studio or other command line T-SQL execution tools such as SQLCMD, OSQL, etc.
Analysis Services	XMLA; the generated XMLA script can be executed in Management Studio.
Reporting Services	VBScript; the generated script cannot be executed in Management Studio, but you can use any script runtime such as CScript or Visual Studio to run and debug the script.

Server Type	Script Language and Execution Environment
Integration Services	None; Integration Services dialogs are not scriptable.
Notification Services	None; Integration Services dialogs are not scriptable.
SQL Server Mobile	None; SQL Server Mobile dialogs are not scriptable.

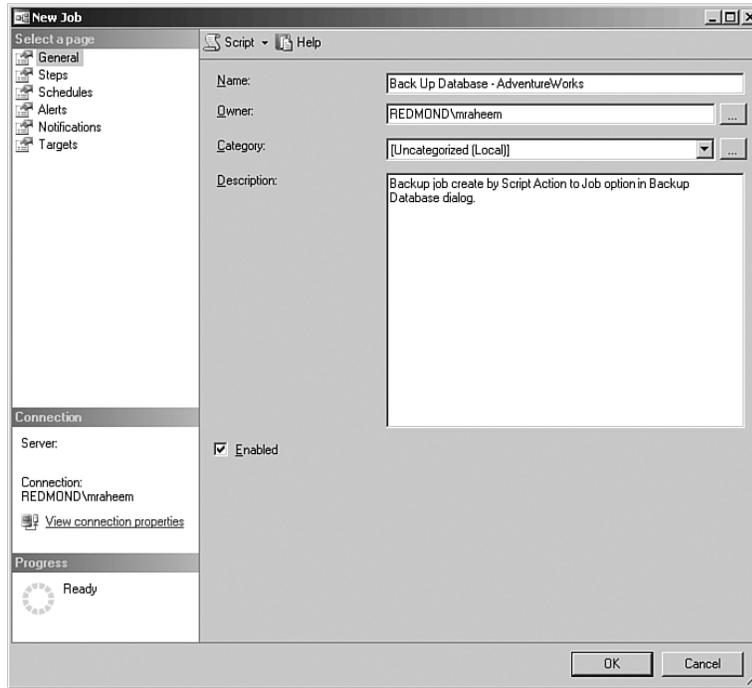
The scripting options are useful for situations in which you want to review and perhaps tweak the script that the management dialog is about to execute on the server prior to its execution. Another usage scenario would be to use the scripting option of the dialogs as a template generator for scripts with long and difficult-to-remember syntax, such as the Analysis Services XMLA.

#### Schedulable

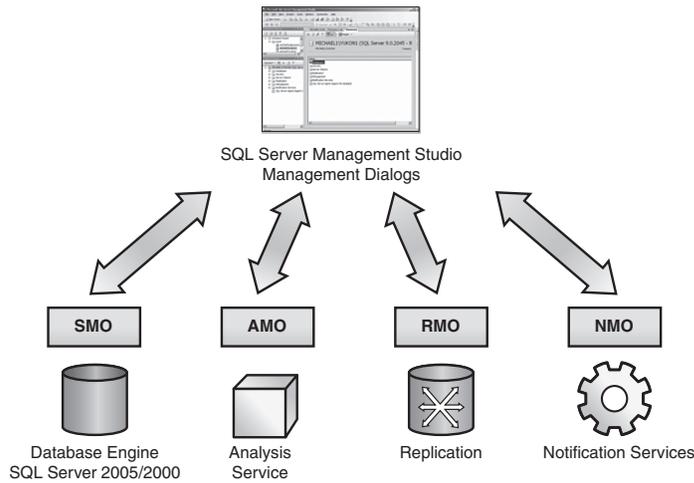
The action of the management dialogs can be scripted as SQL Server Agent jobs to run at a later time or on a recurring schedule. This is particularly useful, for example, in situations in which you want to create a recurring job to back up a database or reorganize an index on the production server at a later time, when the server is not so busy. To create a job for the dialog action, click on the **Script** button on the dialog toolbar and select **Script Action to Job**. On the **New Job** dialog (See Figure 4-15), switch to the **Schedules** page and select or create a new schedule for the job.

Architecturally, all management dialogs are inherited from a base class. This ensures consistency in the layout, formatting, and behavior of the dialogs. In addition, almost all management dialogs use an object model to create/manipulate and perform actions on the server (see Figure 4-16). As a matter of fact, the Microsoft SQL Server team enforced a development policy that all management dialogs must use the object models rather than make a direct connection and embed metadata queries in the dialogs for creation of or performing actions on the server. Enforcing this level of abstraction has been a huge win for SQL Server because it ensured quality in the management tools and reinforced the versatility and flexibility of SQL Server management object models.

## 148 Chapter 4 Inside Management and Administration Tools



**Figure 4-15** The management dialog toolbar provides various scripting options.



**Figure 4-16** Management dialogs leverage SQL Server object models to perform actions on Database Engine, Analysis Services, Replication, SQL Server Agent, and Notification Services.

### **Wizards**

A wizard is a graphical user interface element similar to a management dialog that guides you step by step through a set of complex tasks. Management Studio has a number of wizards to assist the user with management tasks. Similar to the management dialogs, all wizards are inherited from the same class to provide a consistent user experience. Table 4-3 provides a list of the SQL Server Database Engine wizards and a brief explanation of their functionality.

**Table 4-3** SQL Server Database Engine Wizards.

<b>Wizard Name</b>	<b>Functionality</b>
Generate SQL Server Scripts Wizard	Creates T-SQL scripts for all or a subset of objects in a database.
Import and Export Data Wizard	Allows copying, transformation, and loading of data between supported data sources and the Database Engine.
Copy Database Wizard	Provides the functionality to move or copy one or more databases from a source to a target instance.
Maintenance Plan Wizard	Creates scheduled jobs to perform routine database tasks such as back up, update statistics, check integrity, and so on.
Database Mail Wizard	Allows management and configuration of Database Mail.
Configure Distribution Wizard	Configures a server to be a replication distributor.
New Publication Wizard	Provides the functionality to choose objects to replicate.
New Subscription Wizard	Create one or more subscriptions to a publication for replication purposes.

### **Query Editors**

SQL Server Management Studio hosts a number of rich code editors that provide the functionality to create, edit, and execute script-language queries against SQL Server Database Engine, Analysis Services (including

## 150 Chapter 4 Inside Management and Administration Tools

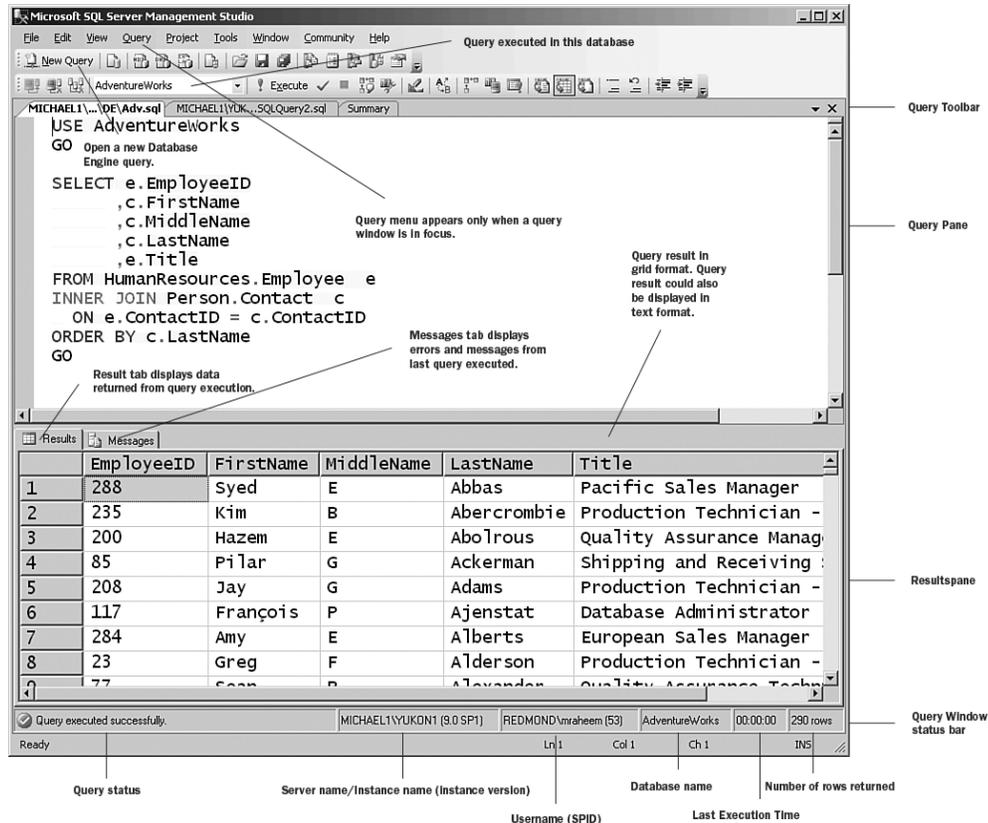
Data Mining), and SQL Server Mobile. Table 4-4 shows which editors work with which server type and through which script language. In addition, Management Studio hosts an XML editor for editing XML files and viewing XML data in query resultsets.

**Table 4-4** Query Editors in Management Studio by Server Type and Script Language

Server Type	Script Language	Query Editor in Management Studio
Database Engine	Transact-SQL (T-SQL)	T-SQL Query Editor
Analysis Services	Multidimensional Expression (MDX)	MDX Query Editor
	Data Mining Extensions (DMX)	DMX Query Editor
	XML for Analysis (XMLA)	XMLA Query Editor
SQL Server Mobile	Transact-SQL (T-SQL)	.SQL Server Mobile Query Editor

Figure 4-17 illustrates various components of a query window. The query pane is where you type a query. The database drop-down in the toolbar provides the functionality to specify the database where you want to execute the query. To execute the query, you can click on the **Execute** button on the query tool bar, select **Query, Execute** from Management Studio main menu, or press **F5**, **Ctrl+E**, or **Alt+X** on the keyboard. You also have the option to execute part of the query by selecting the script and clicking on the Execute button (or pressing any of the aforementioned keyboard shortcuts). The result of the query execution appears in the Results pane. The Results pane has tabs for displaying various components of a query result: Data appears in the **Results** tab, messages are displayed in **Messages** tab, query execution plan is displayed in **Execution plan** tab, and client statistics appear in the **Client Statistics** tab. The data for query results can be displayed in a grid or as text, or it can be redirected to a file. These options can be selected from **Query, Result to** in the main menu of Management Studio, or you can press **Ctrl+T**, **Ctrl+D**, and **Ctrl+Shift+F** for **Results to Grid**, **Results to Text**, and **Results to File**. You can also access these

options from buttons  on the query tool bar. You can hide and show the Results pane by selecting **Window, Hide Result Pane** from Management Studio main menu or by pressing **Ctrl+R** on the keyboard to give you more screen space for editing the query.



**FIGURE 4-17** Management Studio Query Editor and its components.

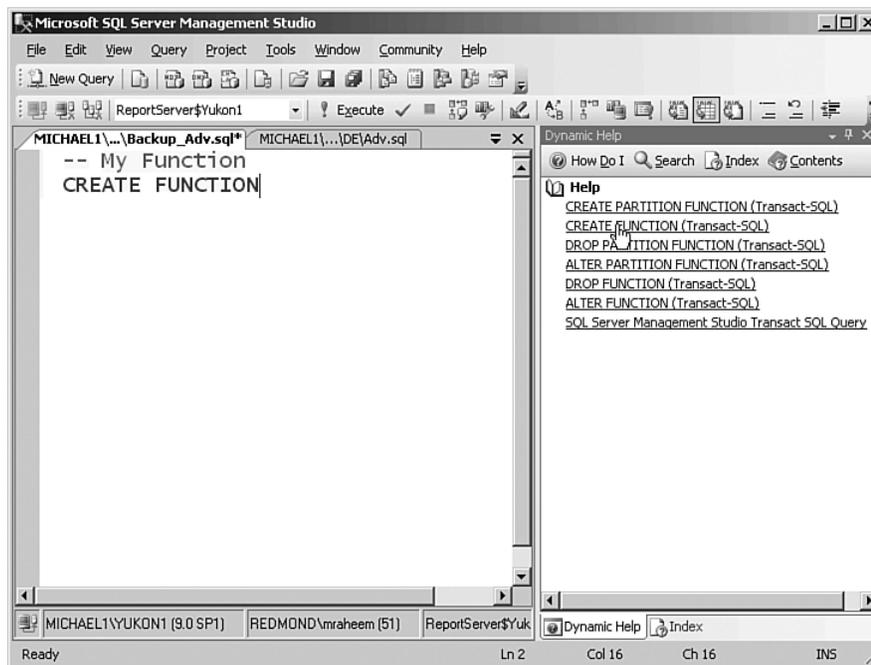
As shown in Figure 4-17, the query window has a status bar on the bottom of the window that displays useful information about the status of the last executed query.

### **Help and Dynamic Help Window**

The Dynamic Help window in Management Studio displays links to relevant help topics while you are performing a task. This is particularly

## 152 Chapter 4 Inside Management and Administration Tools

useful in query editors, where the dynamic help window displays help links as you are typing a query. You can click on a help link to launch Books Online in the context of the help topics as shown in Figure 4-18. This is particularly useful in situations where you want to look up the syntax in Books Online quickly. Alternatively, you can highlight a statement and click F1 to launch help in the context of the statement.



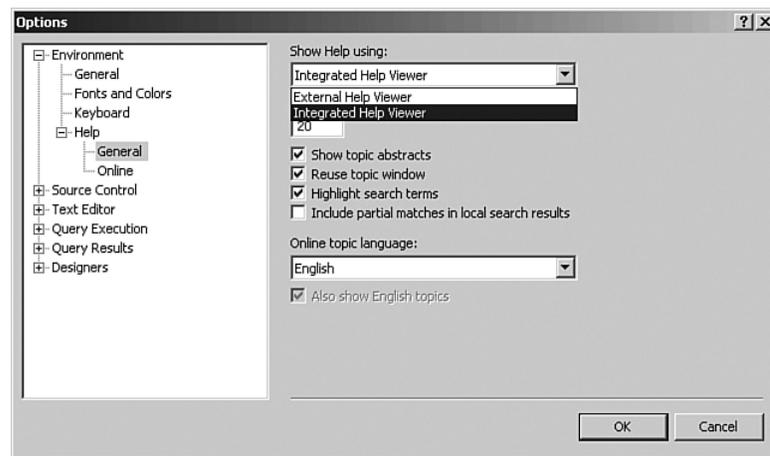
**Figure 4-18** The Dynamic Help window automatically displays links to help topics relevant to the statements in the Query Editor window.

To open the Dynamic Help window, select **Help, Dynamic Help** from the main menu in Management Studio.

To launch Help from Management Studio, press F1 at any time. By default, Help opens as an external window to the Management Studio shell. However, the help windows are still associated with Management Studio, so when you close Management Studio, the help windows close as well. You can optionally open Help as a document window inside Management Studio.

You can configure Help and SQL Server Books Online to launch as document windows inside Management Studio window by following these steps:

1. On the **Tools** menu in Management Studio, select **Options**.
2. In the **Options** dialog, expand **Environment**, expand **Help**, and click on **General**.
3. In the left pane, click the **Show Help Using** drop-down arrow, select **Integrated Help Viewer** (see Figure 4-19), and click **OK**.



**Figure 4-19** The Integrated Help Viewer option will set Help and SQL Server Books Online as document windows inside Management Studio.

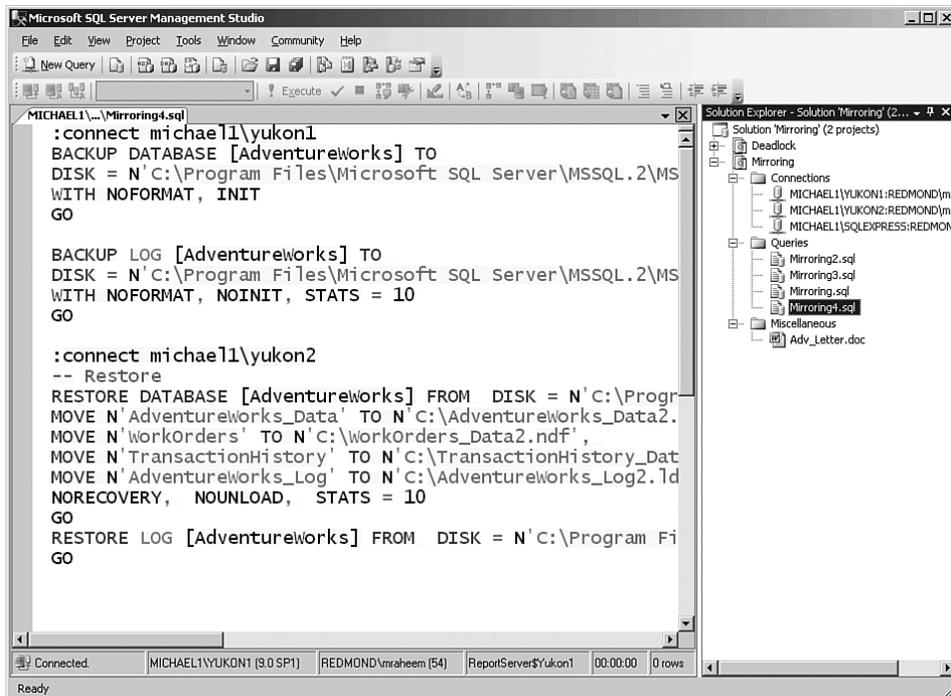
You may need to restart Management Studio for the new setting to take effect.

### **Solution Explorer**

Solution Explorer is a component of Microsoft SQL Server Management Studio that enables you to view and manage project items and perform item management tasks in a solution or a project (see Figure 4-20). It also enables you to use the SQL Server Management Studio editors to work on items associated with one of the script projects. Solution Explorer provides an interface with version control software and the functionality to check in and check out scripts.

## 154 Chapter 4 Inside Management and Administration Tools

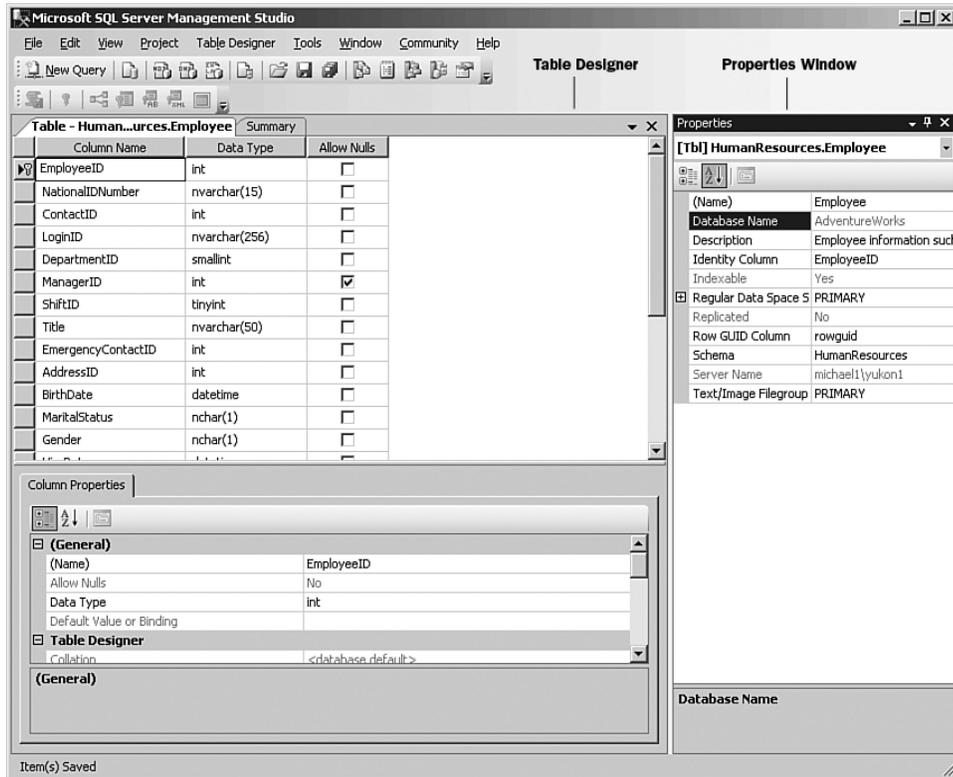
To view Solution Explorer, you can press **Ctrl+Alt+L** on the keyboard, or select **View, Solution Explorer** from the main menu in Management Studio.



**Figure 4-20** Solution Explorer in Management Studio.

### **Properties Window**

The Properties window (see Figure 4-21) provides the functionality to view and change properties of a selected object that are located in editors and designers. By default, the **Properties Window** is located in the bottom-right corner of Management Studio. By default, the Registered Server window appears in the top-left corner of Management Studio. You can easily move, hide, or autohide the window to allocate more space to other components. To quickly view the **Properties** window, click **F4** or choose **View, Properties Window** from the main menu in Management Studio.



**Figure 4-21** The Properties window supports changing properties of objects that are located in editors and designers.

### **Properties window**

Each field in the Properties window displays different types of editing fields, depending on the particular property. Properties shown in grey are read-only.

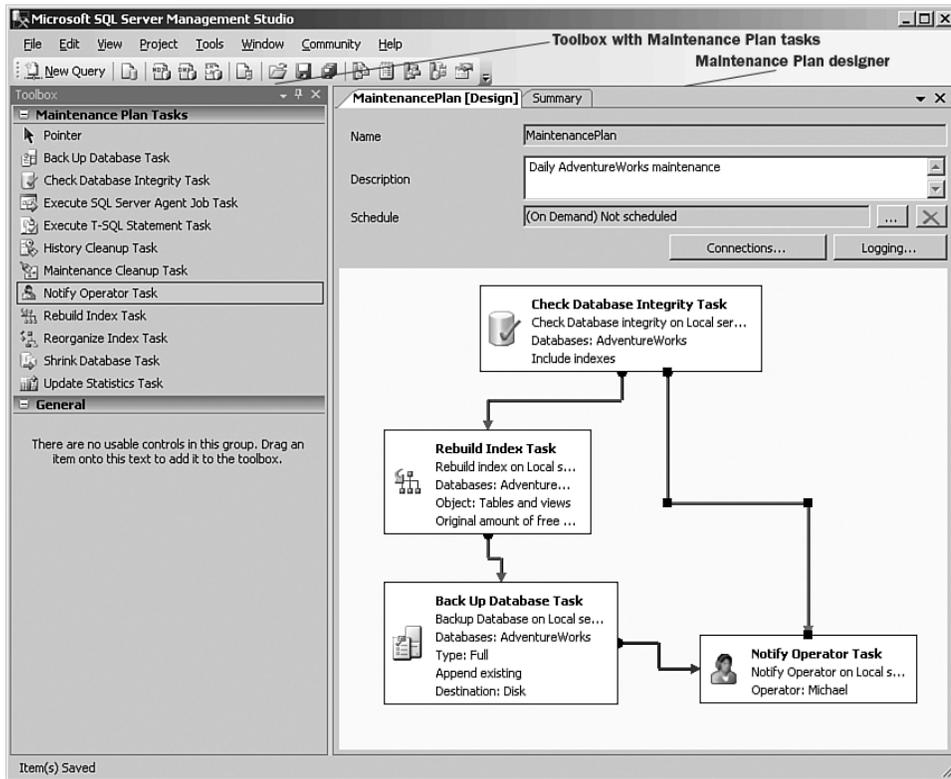
#### **NOTE**

Do not confuse the Properties window with Properties dialogs. Properties dialogs are rich management dialogs that provide the functionality to modify properties of an object on the server. To launch Properties dialogs, select **Properties** from the Object Explorer context menu.

## 156 Chapter 4 Inside Management and Administration Tools

### Toolbox

Toolbox displays objects that can be dragged and dropped into designers and editors. In Management Studio, Toolbox is empty, and it is populated only when the Maintenance Plan Designer is in focus. In Figure 4-22, Toolbox displays a number of Maintenance Plan tasks that can be dragged and dropped into the Maintenance Plan Designer.

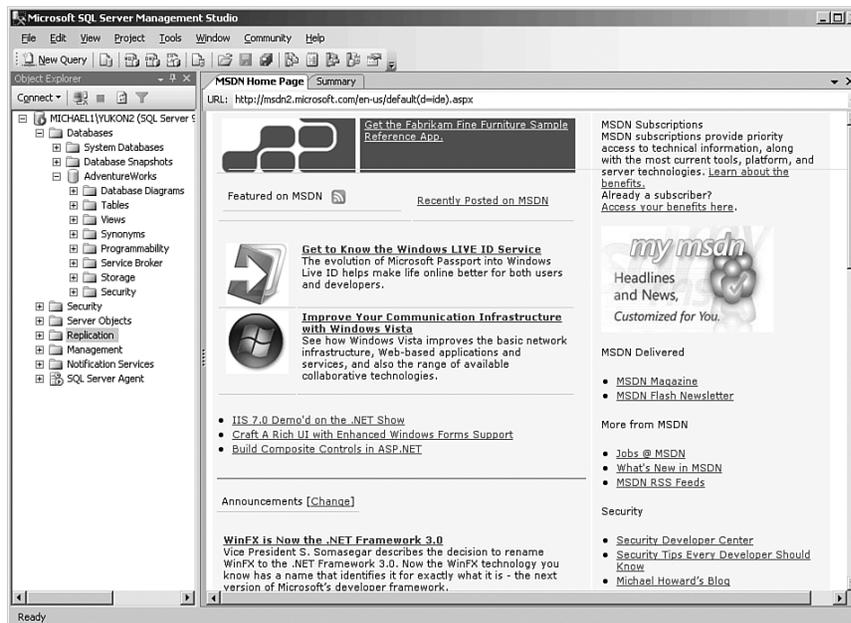


**Figure 4-22** Toolbox is populated with Maintenance Plan tasks only when Maintenance Plan designer is in focus.

### Web Browser

SQL Server Management Studio has an embedded Internet Explorer, which enables you to browse web pages in a document window (see Figure 4-23). This is particularly useful because you can search external resources such as the Microsoft Developer Network ([www.msdn.com](http://www.msdn.com))

or other community web pages without leaving SQL Server Management Studio. You can open the Web Browser window from the main menu in Management Studio. In the main menu, point to **View, Web Browser** and then select **Home** or **Search**. Alternatively, you can open the browser window by selecting **Community, Developer Center** in the main menu.



**Figure 4-23** The embedded browser in Management Studio provides web browsing functionality without the need to leave the environment.

## Management and Administration Scenarios

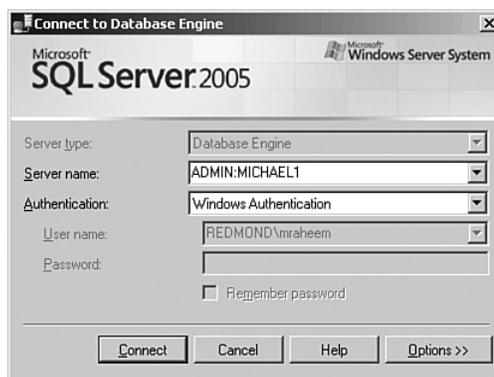
In this section, we originally started writing about the typical management scenarios such as how to back up and restore a database, how to manage indexes, or how to administer security. However, we soon realized that we were just reiterating the scenarios that are already described in product documentation. Instead, we decided to focus on describing real-life (and some painful) scenarios that customers have usually asked us about in person, via newsgroups, or during our presentations. We will

## 158 Chapter 4 Inside Management and Administration Tools

also be describing some interesting scenarios and some features that are not fully supported by Microsoft. Please be cautious in using the undocumented features. We highly recommend that you try to test these results in your development/testing environment before applying them to your production servers.

### Connecting to a Server via Dedicated Administrator Connection

Dedicated Administrator Connection (DAC) is a high-priority connection to the server. DAC is a powerful connection, and it should be used only for troubleshooting, when the server is “hung” and does not accept new connections. SQL Server officially supports DAC connections only from the SQLCMD command line utility tool. However, you can also use Query Editor in Management Studio for DAC connections. To make a DAC connection, prefix the name of the server with **ADMIN:** in the connection dialog. For example, if you want to make a DAC connection to the default instance of Database Engine on a server named **MICHAEL1**, you enter **ADMIN:MICHAEL1** in the **Server name** field on the connection dialog (see Figure 4-24). It is good practice to disconnect and free up the DAC connection as soon as your troubleshooting is complete.



**Figure 4-24** Making a DAC connection from Management Studio.

DAC is limited to only one connection. If you use DAC for a query window and try to open other components of Management Studio such as Object Explorer in the context of the same connection, you will get strange errors. This is one of the reasons Microsoft does not support making DAC connections with Management Studio.

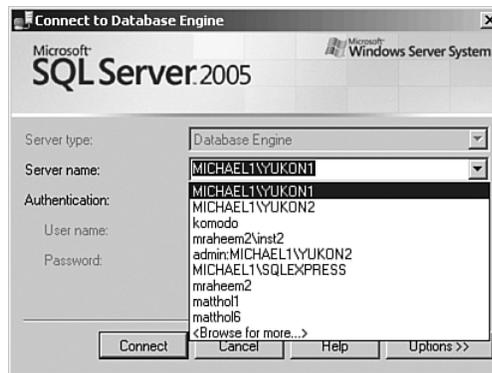
**NOTE**

By default, you cannot make a remote DAC connection. Remote DAC is disabled on new installations of SQL Server. See the topic “Remote DAC” in Chapter 3 for information about enabling this option.

### Removing Most Recently Used (MRU) Server Names from the Connection Dialog

The **Server name** drop-down list box in the Connection dialog stores the 10 most recently used (MRU) server names (see Figure 4-25). The Connection dialog stores the server name after it has successfully connected to the server. Although this is a convenient feature, there are situations in which you may want to remove the dormant servers from this list or remove all the entries for security reasons. You may also want to copy the list to another workstation or share it with another user.

The server name MRU list is distinct for each user on a computer. The list is stored in a file named **mru.dat** under the **Application Data** directory for each user:



**Figure 4-25** Making a DAC connection from Management Studio.

```
C:\Documents and Settings\\Application Data\
Microsoft\Microsoft SQL Server\90\Tools\Shell\mru.dat
```

To remove all server entries from the list, delete or rename the file to oldmru.dat. To share the MRU list with another user, identify the user's mru.dat file, and replace it with the source mru.dat file. To copy the

## 160 Chapter 4 Inside Management and Administration Tools

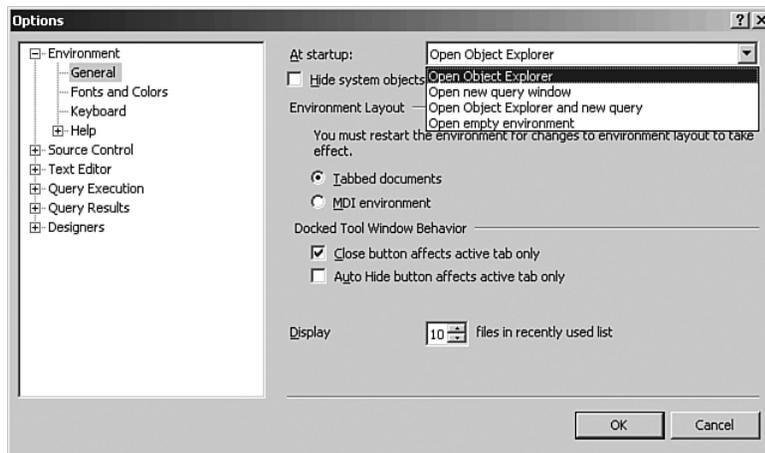
connection dialog server MRU list to another server, simply copy the source mru.dat file from the source to the target server.

### Changing Default Startup Behavior of Management Studio

By default, launching Management Studio displays the connection dialog, prompting you for a connection to a server. After it is connected, it opens Object Explorer in the context of the specified server. You have the option to change this default behavior to opening a new query window, opening Object Explorer and a new query window, or not being prompted for a connection at all.

To change the startup behavior of Management Studio, do the following:

1. Click on the **Tools** menu in Management Studio and select **Options**.
2. In the **Options** dialog, expand **Environment** and select **General**.
3. In the **At Startup** drop-down, select the desired startup behavior (see Figure 4-26). Click **OK**.

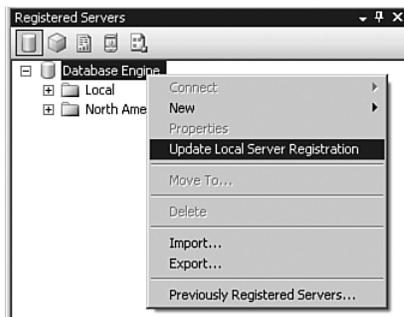


**Figure 4-26** You can change the startup behavior of Management Studio to better suit your everyday tasks.

You may need to restart Management Studio for the new setting to take effect.

## Updating Registered Servers with Local Instances

When a user launches SQL Server Management Studio for the first time, it discovers the local instances of SQL Server and automatically adds them to the list of registered servers for that user. However, if new instances are added or old instances are removed, the registered servers could get out of synch with the actual instances. Of course, the user could add or delete instances manually (see Figure 4-27), but there is an automated way to register the local instances quickly.



**Figure 4-27** Updating Registered Servers list with local instances of SQL Server.

To add or update the list of your registered servers automatically with local instances of SQL Server on a computer, right-click on the root node of a server type (such as Database Engine), and select **Update Local Server Registration**. Be aware that this option works only for the selected server type. To update all server types, you have to select the server type from the Registered Server toolbar and perform an update.

## Importing Registered Servers from Enterprise Manager

SQL Server 2005 provides the functionality to keep your SQL Server 2000 registered servers during both upgrade and migration to SQL Server 2005.

If you upgrade an instance of SQL Server 2000 to SQL Server 2005 by using the SQL Server 2005 setup program, the old Registered Servers in Enterprise Manager are automatically upgraded to registered servers in Management Studio. However, if you do a side-by-side installation of SQL Server 2005 on the same computer as SQL Server 2000, Enterprise Manager and Management Studio do not share the same registered servers. In this case, you can optionally import the SQL Server 2000 Enterprise Manager registered servers to Management

## 162 Chapter 4 Inside Management and Administration Tools

Studio. To do this, right-click on the **Database Engine** node in the Registered Servers window in Management Studio and select **Previously Registered Servers** (See Figure 4-28).



**Figure 4-28** Migrating registered servers from Enterprise Manager.

### Referencing Registered Servers with a Friendly Name

Server names do not usually correspond to the functionality of or the role of the server. In addition, some server names may be too long or too similar to one another, which makes you more prone to performing the wrong task on the wrong server. Management Studio provides the capability to register a server with a friendly name.

Figure 4-29 displays the **New Server Registration** dialog. The name shown in the field **Server name** is the physical name of the server, but the one shown in the field **Registered server name** could be the same or different from the physical name. By default, **Registered server name** is the same as **Server name**. As you type a name in the **Server name** field, the same name is automatically entered in **Registered server name**. However, after entering **server name** you can override the **Registered server name** with a friendly name. In Figure 4-29, the physical name of the server is **SRVBCVANDB2005-903ha**, but its registered server name is **QA Server located in South Vancouver**.

As shown in Figure 4-30, registered servers are displayed by registered server name, followed by server name in parentheses. In the example, it is displayed as **QA Server located in South Vancouver (SRVBCVANDB2005-903ha)**.

The preceding paragraphs described how to register a new server with a friendly name. You can also assign a friendly name to a previously registered server by editing its properties. To do so, right-click on the

Registered Server window and select **Properties** in the exposed context menu. This launches the **Edit Server Registration Properties** dialog, where you can change the value of the **Registered server name** field.



**Figure 4-29** You can assign a friendly name to your registered servers.



**Figure 4-30** Registered Server window displays the friendly names.

## Copying Registered Servers Between Users and Computers

Registered servers are stored on the local computer where Management Studio is running. The registered server information is stored for each user and it is not shared between users. To copy the registered server information between two computers or two users, you need to use the import and export functionality of registered servers.

**164 Chapter 4 Inside Management and Administration Tools**

You can follow these steps to export the registered server information for a user on a particular computer:

1. Log in as the user to the computer that contains the registered server information.
2. Launch SQL Server Management Studio.
3. From the main menu, select **View, Registered Servers**.
4. From the **Registered Server** toolbar, select a server type, such as **Database Engine**.
5. In the Registered Server window, right-click and select **Export** from the context menu. This launches the Export Registered Servers dialog, as shown in Figure 4-31.
6. In the Export Registered Servers dialog, you have the option of exporting all or just a subset of the registered servers. To export all registered servers for the selected server type, select the root node (e.g., **Database Engine** in Figure 4-31). To export a subset of the servers, select the server group (e.g., **Canada** in Figure 4-31).
7. Specify the **Export file** by either typing the path and name of the file or choosing the browse option next to it. Click **OK**.



**Figure 4-31** Management Studio provides the functionality to export registered servers.

**NOTE**

The option **Do not include user names and password in the export** file is applicable only to registered servers that used the SQL Server Authentication option and specified a password during registration. For security reasons, we highly recommend you do not include the passwords in the export file and let the user who imports the file specify the passwords after importing the file.

To import the registered server information for a user on the same or a different computer, follow these steps:

1. Follow the same export steps 1–4 as just described.
2. In the Registered Server window, right-click and select **Import** from the context menu. This launches the Import Registered Servers dialog, as shown in Figure 4-31.
3. Specify the **Import file** by either typing the path and name of the file or choosing the browse option next to it.
4. Select the server group to import the file (e.g., **North America** in Figure 4-32). Click **OK**.



**Figure 4-32** Management Studio provides the functionality to import the registered server's information from a previously exported file.

**NOTE**

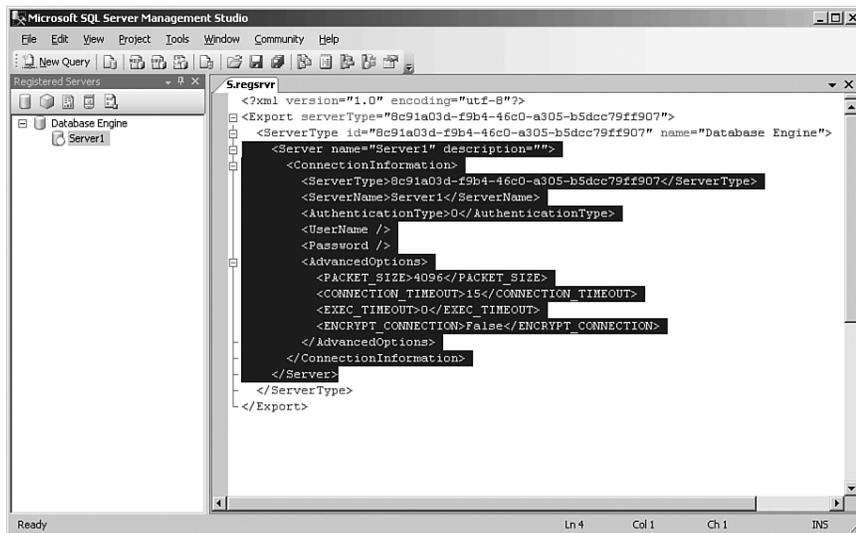
It is good security practice to delete the registered servers import file after you have completed the import operation.

## Adding Registered Servers to an Export File

As you have seen above, Management Studio provides a user interface to add registered servers. But what if you are in a situation where you have to create tens or hundreds of registered servers? The answer is to leverage the SQL Management Objects (SMO). SMO's `Microsoft.SqlServer.Management.Smo.RegisteredServers` namespace provides a rich library of classes that you can use in your Visual Basic .NET or C# code to create, remove, and manage registered servers. You can look up the library in Books Online.

However, there is an undocumented and perhaps easier way to manipulate the registered servers' information without using SMO. The registered servers export file is in XML format. You can append the new registered servers to the XML file.

To understand the schema of the export file, try creating a registered server named **Server1** and export the registered server to a file named `s.regsrv`. Next, open `s.regsrv` in the XML editor in Management Studio. As shown in Figure 4-33, you can see that the server information is surrounded by `<Server>` tags.



**Figure 4-33** Editing a registered server export file in the XML editor in Management Studio.

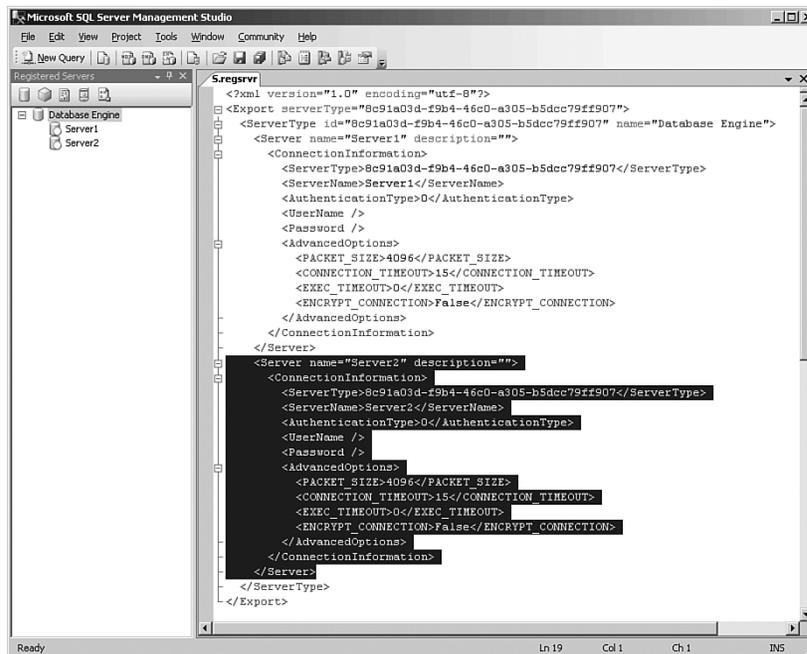
To add a server named **Server2** to the export file, copy and paste the text between server tags. Next, in the pasted text change **Server1** to **Server2** (indicated in bold characters as follows):

```

<Server name="Server2" description="">
  <ConnectionInformation>
    <ServerType>8c91a03d-f9b4-46c0-a305-b5dcc79ff907</ServerType>
    <ServerName>Server2</ServerName>
    <AuthenticationType>0</AuthenticationType>
    <UserName />
    <Password />
    <AdvancedOptions>
      <PACKET_SIZE>4096</PACKET_SIZE>
      <CONNECTION_TIMEOUT>15</CONNECTION_TIMEOUT>
      <EXEC_TIMEOUT>0</EXEC_TIMEOUT>
      <ENCRYPT_CONNECTION>False</ENCRYPT_CONNECTION>
    </AdvancedOptions>
  </ConnectionInformation>
</Server>

```

Next, save the `s.regsvr` export file and try importing it in Management Studio. You will see `Server1` and `Server2` in the Registered Servers window, as shown in Figure 4-34.

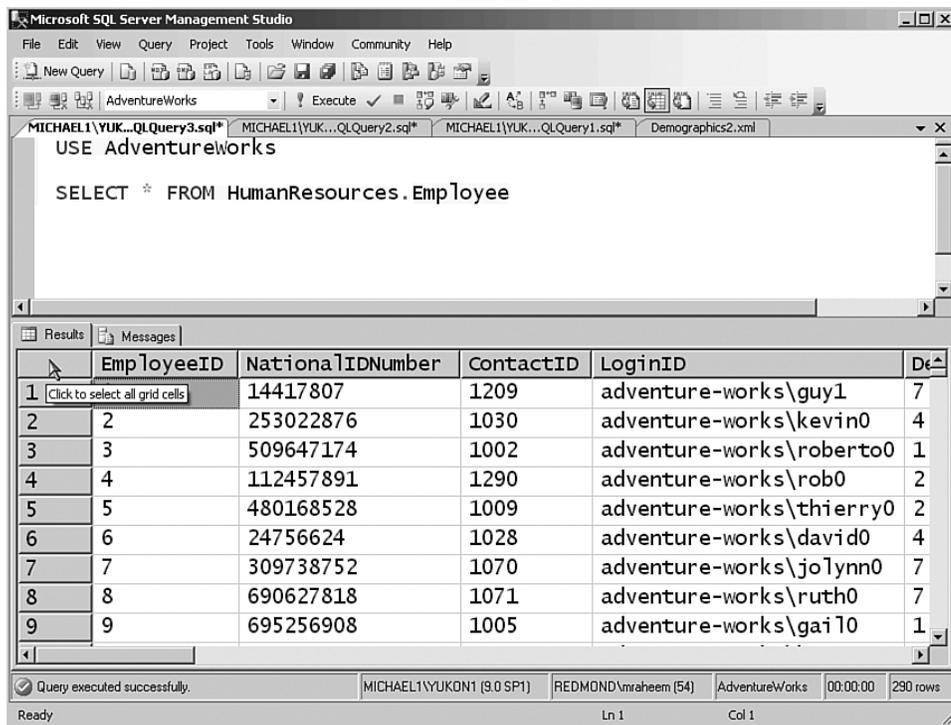


**Figure 4-34** Manipulating the registered server export file to add a new registered server.

## 168 Chapter 4 Inside Management and Administration Tools

### Copying Query Result to Excel Spreadsheet

You can copy part or all of a query result to a Microsoft Office Excel spreadsheet or any other text editors. To export all the resultset, right-click on the results grid and select **Save Results As** in the context menu. This option saves the results in a Comma-Separated Value (CSV) file, which you can subsequently open in Excel. Alternatively, you can select all or part of the data in the results grid and copy and paste it in an Excel spreadsheet. To select all the cells in the result set grid, you can right-click and choose **Select All**, or click on the top-left corner of the results grid as displayed in Figure 4-35.

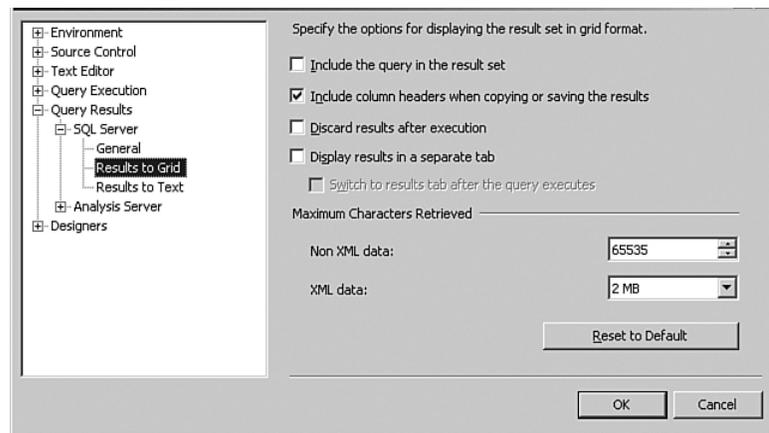


**Figure 4-35** You can click on the top-left corner of the result grid to select all data in a query result set.

If you followed these instructions, you may have noticed that neither saving the results to a CSV file or copying all or part of the result set includes the column headers. To include the column headers in the file or copied data, follow these steps:

1. Click on the **Tools** menu in Management Studio and select **Options**.
2. In the Options dialog, expand **Query Results**, and **Results to Grid**. Select **Results to Grid** (see Figure 4-36).
3. Select **Include column headers when copying or saving the results**. Click OK.

You may need to re-execute your query in a new query window for the new setting to take effect.



**Figure 4-36** You can choose the option to include column headers when copying a resultset to Excel.

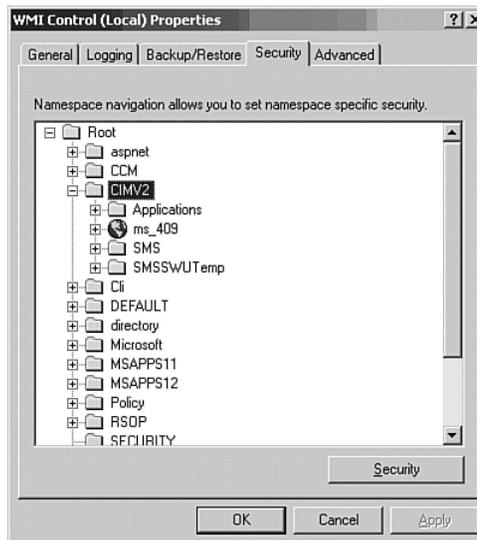
## Configuring WMI to Show Server Status in Registered Servers

Registered Servers and Object Explorer windows display the state of SQL Server services through a small icon appearing to the left of the server node. Both Registered Servers and Object Explorer use Windows Management Instrumentation (WMI) to query the state of the services. To query the status of the services successfully, two things must happen: 1) The computer that hosts the SQL Server services must have WMI installed, and 2) the user who is running Management Studio must have permissions to access the WMI object remotely on the computer that hosts the SQL Server services.

You can follow these steps to grant WMI permission to a particular user.

**170 Chapter 4 Inside Management and Administration Tools**

1. Log on to the computer that is hosting the SQL Server services.
2. On the Windows Start menu, select **Run**, type **mmc**, and click **OK**.
3. From the **File** menu in the **MMC** console, select **Add/Remove Snap-in**, and then click **Add**.
4. In the **Add Standalone Snap-in** dialog, select **WMI Control**, and click **Add**.
5. On the Change Managed Computer dialog, select **Local computer** and click **Finish**.
6. In the Add Standalone Snap-in dialog, click **Close**.
7. In the Add/Remove Snap-in dialog, click **OK**.
8. In the **MMC** console, right-click on **WMI Control (Local)** and select **Properties**.
9. In the WMI Control (Local) Properties dialog, select the **Security** tab, expand the **Root** node, and select **CIMV2** (see Figure 4-37).



**Figure 4-37** To view the state of SQL Server services, the user must have permissions to the WMI CIMV2 namespace.

10. Click on **Security** at the bottom of the dialog.
11. Add the user or a group of which the user is a member to the **Group or user names** box and select it.

12. In the **Permissions for <group or user>** box, select the **Allow** column for the **Remote Enable** permission and click **OK**.
13. Close the MMC console.

## Leveraging an Asynchronous Population of Objects in Object Explorer

You can take advantage of the asynchronous operation of Object Explorer, especially when you are dealing with a large number of server objects. This means that while Object Explorer is busy populating nodes, Management Studio remains responsive and you can continue performing other tasks such as editing or executing a query, expanding another node in Object Explorer, connecting Object Explorer to another server, and even canceling the population of the nodes. The capability to cancel is a great feature for situations in which you have accidentally expanded a highly populated node and you want to cancel the population and apply a filter instead.

To better understand the asynchronous operation of Object Explorer, make use of the following script. First, you need to create a highly populated node in Object Explorer. You can execute the following script below to create 25,000 tables in the AdventureWorks database.

```
-- This script creates 25,000 tables in AdventureWorks database
USE AdventureWorks
GO

DECLARE @i          INT
        ,@table_ddl VARCHAR(4000)

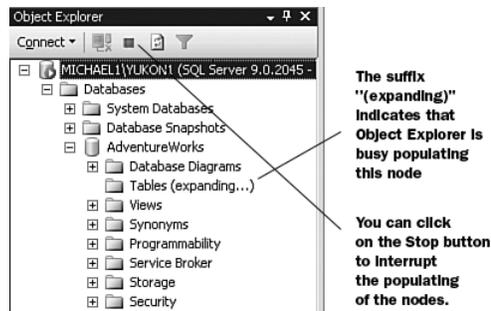
SET @i = 100001

WHILE @i - 100000 <= 25000
BEGIN
    SET @table_ddl = 'CREATE TABLE Table'
                  + SUBSTRING(CONVERT(CHAR, @i), 2, 5)
                  + '(ID INT NULL)'
    EXECUTE(@table_ddl)
    SET @i = @i + 1
END
```

After the script has executed successfully, try connecting Object Explorer to the server and expanding the Tables node. Notice that while Object Explorer is busy populating the Tables node, the phrase (expanding...) is appended to the end of Tables node (see Figure 4-38). Now you can

## 172 Chapter 4 Inside Management and Administration Tools

collapse the Tables node and perform other tasks in Management Studio. You can even click on the Stop button in the Object Explorer toolbar to stop populating the nodes.



**Figure 4-38** Asynchronous operation of Object Explorer provides the functionality to stop the population or perform other tasks during the population of the node.

The asynchronous behavior of Object Explorer is due to its multi-threaded operation. Object Explorer can spawn up to four threads for its population of the nodes. This means you expand up to four highly populated nodes in Object Explorer and continue performing other tasks in Management Studio.

### Editing Queries in Disconnected Mode

Unlike Query Analyzer in SQL Server 2000, Management Studio enables you to edit queries without connecting to a server. This is particularly useful in situations in which the server is not available and you still want to take advantage of the rich editor features such as keyword color-coding, dynamic help, or source control. Another potential situation is one in which you want to guard yourself against accidental execution of an incomplete script against the server.

If you want to open a new script in disconnected mode, select **File, New, Database Engine Query** (or any other query type). Next, click on **Cancel** in the Connection dialog. The editor is now ready to type a new query. Notice the title of the query says **not connected** and the query window status bar also says **Disconnected**. As soon as you are ready to execute the query, you click on the connect button  in the query toolbar or select **Query, Connection, Connect** from the Management Studio main menu.

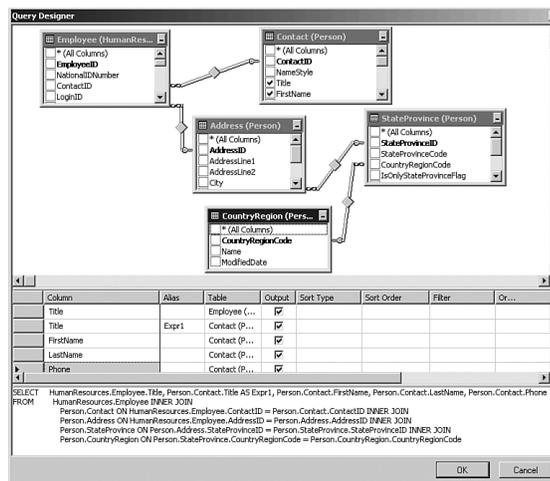
If you want to open an existing query in disconnected mode, simply click on **Cancel** in the connection dialog.

If you have a query that is already connected and you want to continue editing it in a disconnected mode, click on the disconnect button in the Management Studio query toolbar or select **Query, Connection, Disconnect** from the main menu.

To disconnect all queries, select **Query, Connection, Disconnect All Queries** from the main menu in Management Studio.

### Using Query Designer for Writing Queries

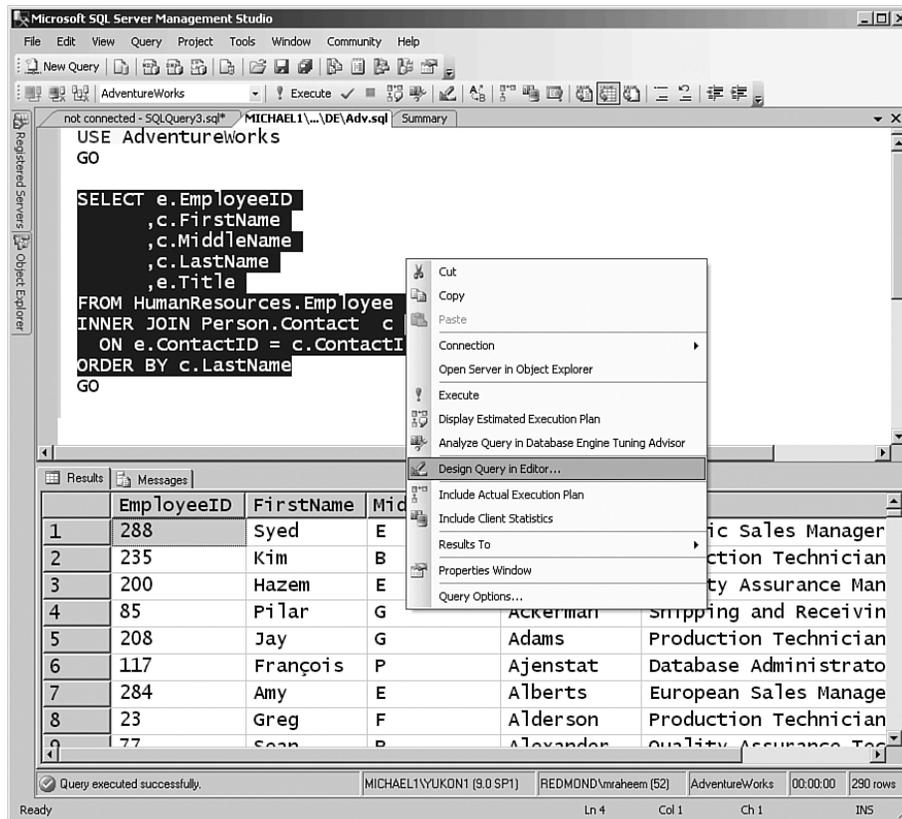
Management Studio provides a graphical query designer for authoring Transact-SQL queries. To design a new query, open a new query window, and press **Ctrl+Shift+Q** on the keyboard or select **Query, and Design Query in Editor** from the main menu in Management Studio. In the **Add Table** dialog, you can select tables, views, functions, or synonyms for your query. The Query Designer automatically discovers the relationship between the selected objects. After you have selected the objects, you can define the output columns, sorting, or grouping in the resultset. While you are designing the query graphically, the SQL pane on the bottom of the query designer displays the T-SQL syntax of your query (see Figure 4-39).



**Figure 4-39** Query Designer in Management Studio provides the functionality to graphically author a Transact-SQL query.

## 174 Chapter 4 Inside Management and Administration Tools

You can also get a graphical representation of an existing query in Query Designer by selecting all or part of the T-SQL script in a query window and selecting the option **Design Query in Editor** from the context menu or the main menu in Management Studio (see Figure 4-40). Any changes to the query in Query Designer updates the query in the query window.



**Figure 4-40** Management Studio provides the functionality to design all or part of a T-SQL query in Query Designer.

### Executing a Query Across Multiple Servers

One of the common management actions is executing a script across multiple servers. The query editors in Management Studio provide a couple of solutions for executing scripts across multiple servers.

The first option is to use the Change Connection feature of query editors in Management Studio. Let's go through an example and see how you can leverage this feature. Suppose your objective is to execute the following T-SQL script to back up the AdventureWorks database across three servers: SERVER1, SERVER2, and SERVER3.

```
-- To permit log backups, before the full backup, alter the database
-- to use the full recovery model
USE master
GO
ALTER DATABASE AdventureWorks
    SET RECOVERY FULL

-- Back up the full AdventureWorks database
BACKUP DATABASE AdventureWorks
    TO DISK = 'C:\BAK\Adv_Full.bak'

-- Back up the AdventureWorks log
BACKUP LOG AdventureWorks
    TO DISK = 'C:\BAK\Adv_Log.bak'
```

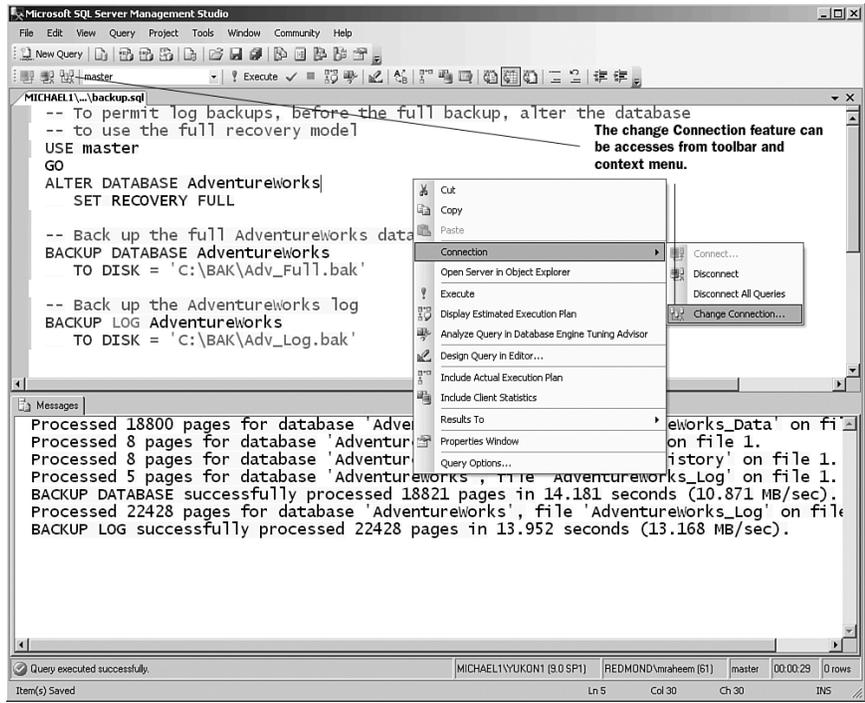
Open up a new database engine query, connect to SERVER1, copy and paste this script, save the query as **backup.sql**, and execute it. To execute the same script on SERVER2, you do not need to close the script and reopen it (as you had to in SQL Server 2000 Query Analyzer). Instead, click on the **Change Connection** button  on the query toolbar, or right-click in the query window and select **Change Connection** from the context menu (see Figure 4-41). This will launch the connection dialog, where you can connect to **SERVER2** and execute the script. You can change the connection again to execute the script on **SERVER3**.

Another option to execute a T-SQL query across multiple servers is to use the SQLCMD Mode feature of the T-SQL query editor. Let's go through the same example and see how you can execute the script across SERVER1, SERVER2, and SERVER3 by using SQLCMD mode.

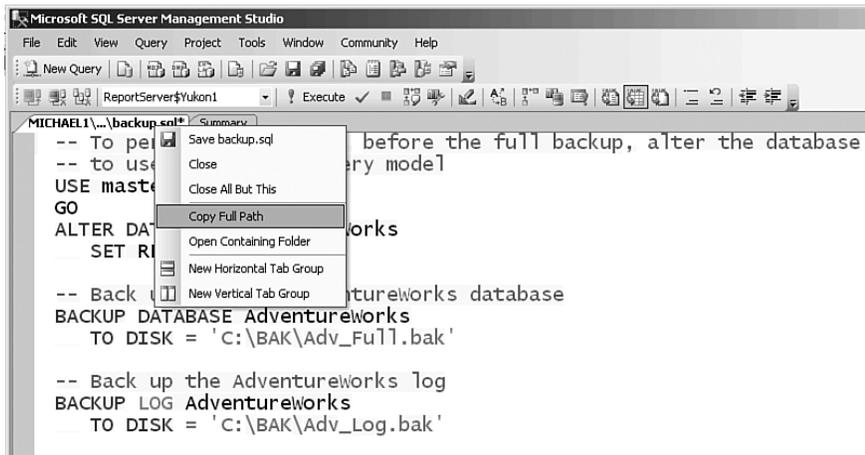
In Management Studio, open the **backup.sql** file that you saved earlier. Right-click on the tab of the query window and select **Copy Full Path**, as shown in Figure 4-42. The full path in the example is C:\BAK\Adv\_Full.bak.

Next, open a new query window, select SQLCMD mode by either clicking on the SQLCMD button  on the query toolbar or selecting **Query, SQLCMD Mode** from the Management Studio main menu. Now, the query is ready to accept SQLCMD extensions. You can use **:connect <servername>** to connect to SERVER1, SERVER2 and

**176 Chapter 4 Inside Management and Administration Tools**

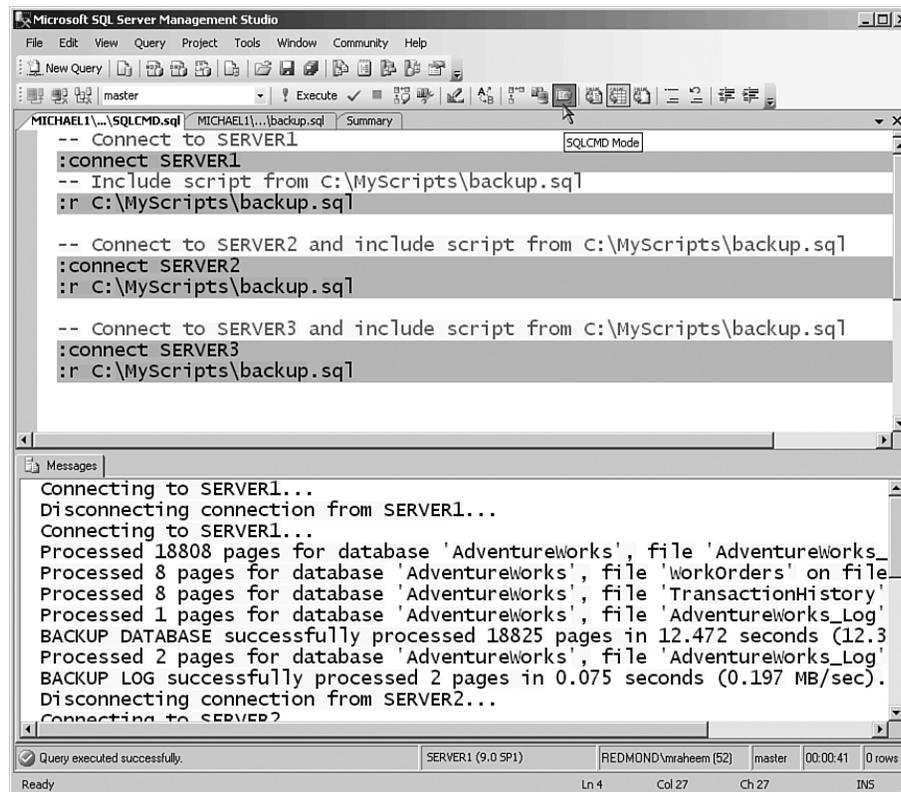


**Figure 4-41** The change connection feature of Management Studio enables you to connect a query to another server.



**Figure 4-42** Context menu of the query window provides a number of features, such as copying the full path of the query.

SERVER3 and use `:r <file path>` to include the backup script. You can paste the copied file path rather typing it. Notice the lines of code that contain SQLCMD extensions appear with a grey background. Your script should look like the script shown in Figure 4-43.



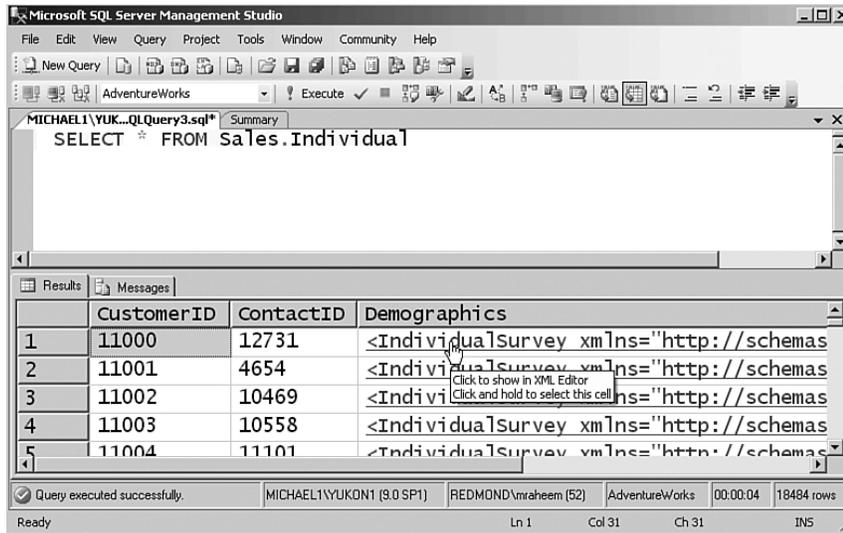
**Figure 4-43** Management Studio provides the functionality to execute queries in SQLCMD mode.

## Viewing and Editing XML Data

Management Studio hosts a rich XML editor. In addition, the XML editor is integrated with T-SQL query editor for viewing XML data. To open an XML file in Management Studio, select **File, Open, File** in the main menu and select the XML file in the Open File dialog.

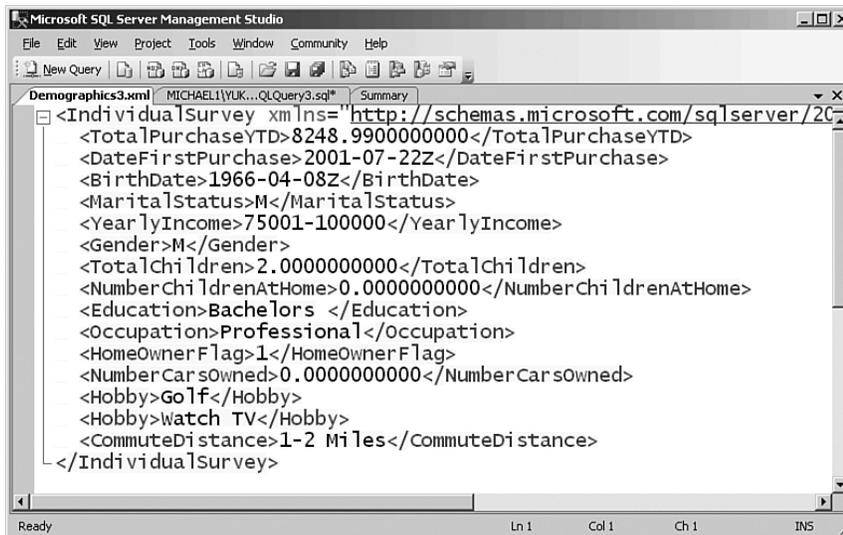
With the XML datatype now being a native datatype in SQL Server, the XML query result is displayed as a link in the grid format (see Figure 4-44).

## 178 Chapter 4 Inside Management and Administration Tools



**Figure 4-44** XML data returned as part of query execution in grid format.

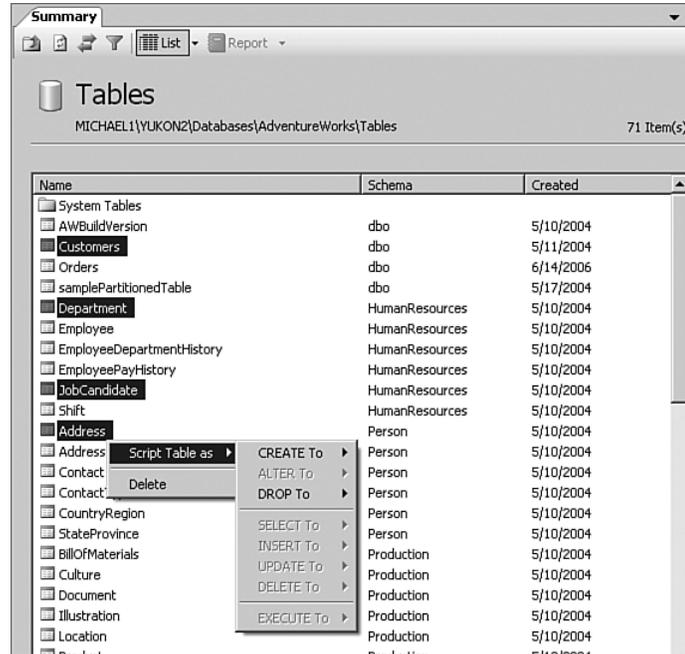
To view the XML resultset in the XML editor, click on the **XML** link in the results grid (see Figure 4-45).



**Figure 4-45** XML query results can be viewed in the XML Editor of Management Studio.

## Deleting and Scripting Multiple Objects

Using Object Explorer, you can delete or generate a script for one object. So how would you go about deleting multiple objects or quickly generating a DDL script for a selection of objects? The answer is in the Summary window. The Summary window allows deletion of multiple objects and the generating of CREATE and DROP scripts. To select a range of nodes, click on the first node, hold the **Shift** key, and click on the second node. To select an individual node, click on the first node, hold the **Ctrl** key, and click on the subsequent nodes. After the nodes are selected, choose the **Script** or **Delete** options in the Summary window context menu (see Figure 4-46).



**Figure 4-46** The Summary window provides the functionality to delete or script multiple objects.

### NOTE

The Summary window provides only simple scripting options. For more advanced scripting options, use the Generate SQL Server Scripts Wizard.

**180 Chapter 4 Inside Management and Administration Tools**

---

**Summary**

---

As you have seen in this chapter, SQL Server Management Studio is a rich and comprehensive tool that enables many management and administration scenarios. We have also given you some tips and tricks for using the tool. However, the scenarios described in this chapter represent just a few of the things you can do with Management Studio. You can find additional usage scenarios in the product documentation and in Books Online.