

SQL Server 2005 Management Tools

Explore the many new management tools for administrators and developers

Kalen Delaney,
Ron Talmage
SQL Server Magazine InstantDoc #46798
August 2005

SQL Server 2005 delivers an entirely new set of management tools for both DBAs and developers. With the addition of some new services in SQL Server 2005, service management is becoming a more important area of DBA concern. The new SQL Server Management Studio replaces both the SQL Server 2000 Enterprise Manager and the Query Analyzer. You need to be aware of two supporting tools: the SQL Server Configuration Manager and the Surface Area Configuration tool. Let's look at each in turn before turning to the SQL Server Management Studio.

Note that SQL Server 2005 is going through its last beta cycle as a series of Community Technology Previews, which are interim beta-level, pre-release builds. Some minor changes in these management tools may occur before release. The information in this article is based on the feature-complete April Community Technology Preview (CTP).

Configuration Management Tools

You'll need to use three new small tools whenever you install SQL Server 2005. For increased security, SQL Server 2005 sets most services and external connectivity features off by default, so if you need anything more than the options you chose during setup, you'll have to use these tools. With the first tool, you can manage SQL Server services; the other two tools form a set that you can use to manage the interfaces through which unintended or malicious access to SQL Server can occur. We call this set of interfaces SQL Server's attack surface area.

SQL Server Configuration Manager. The SQL Server Configuration Manager is a helper tool that you'll need to manage the Windows login accounts for the various SQL Server 2005 services on your server as well as the services themselves. You can't start or stop SQL Server 2005 services directly from SQL Server Management Studio as you can with the Enterprise Manager, so you'll need to learn the SQL Server Configuration Manager right away. You can invoke it from the Windows Start Menu or from the Computer Management applet in the Windows Administrative Tools dialog. If you're running SQL Server Management Studio, which you'll learn about in the next section, you can launch it from the Management Studio Registered Servers dialog.

You use the SQL Server Configuration Manager to manage SQL Server services, network libraries, and the SQL Native Client, as [Figure 1](#) illustrates. Here, you can set and change service accounts and enable and disable services. This tool has some overlap with the next two surface-area configuration tools, which manage these services and other features from a security-threat management standpoint.

Surface-area configuration tools. Many new services are available in SQL Server 2005. In addition to the standard SQL Server, SQL Agent, and Analysis Server services, you can now configure services for Integration Services (formerly Data Transformation Services—DTS), Reporting Server, Full-Text Search, and SQL Browser.

These services are part of the potential attack surface of a SQL Server 2005 installation. To minimize the potential attack surface, Microsoft has provided two new surface-area configuration tools, one for managing services, and another for connectivity features. You'll notice links for these tools at the end of a SQL Server 2005 setup process, but you can also get to them from the Windows Start Menu. Both of these tools help join the disparate services and features that expose a SQL Server to potential attack, and you can enable and disable those services directly through these tools.

The Surface Area Configuration for Services and Connections tool allows you to enable or disable the various services, as shown in [Figure 2](#). You can also disable and enable the SQL Browser service, which listens on UDP port 1434 and handles connections to named instances.

In the Surface Area Configuration for Features tool, which [Figure 3](#) shows, you can enable connectivity features, including ad-hoc remote queries, CLR integration, Database Mail (SMTP) stored procedures, remote Dedicated Administrator Connections, native Web services (SOAP) endpoints, the OLE automation extended stored procedures, Service Broker endpoints, SQL Mail (MAPI) stored procedures, xp_cmdshell, and the Web Assistant. By default, all these features are disabled and must be explicitly enabled. You can also use the sp_configure system stored procedure to enable many of the features and the relevant T-SQL commands to enable the endpoints.

Management Studio

SQL Server 2005 delivers a consolidated management tool with the SQL Server Management Studio, a comprehensive tool that combines features relevant to both DBAs and developers. Management Studio is a complex tool and contains too many features to cover in detail in this article. The major new changes can be organized along three lines: server management, query editing, and performance analysis.

Management Studio is the great "tool consolidator": Microsoft deliberately combined the best administrative features of Enterprise Manager and the best query-editing and analysis features of Query Analyzer into an entirely new tool. At the same time, Management Studio adds many new features not available in either tool. We'll take a closer look at the new features later in the article.

When you first launch Management Studio, you'll notice a Visual Studio look and feel. But even though Management Studio is based on the Visual Studio shell and has some of the same layout, don't let looks fool you: Management Studio is a completely new tool written in managed code.

What you'll notice immediately are two basic types of dialogs: the central, non-movable area of the Management Studio Window is the document window area, and the peripheral areas are called components. By default, the document window initially displays the Summary page, which contains several reporting functions about registered SQL Servers. When you first start up Management Studio, you'll see the Registered Servers and Object Explorer component dialogs on the left side of the window and the Summary page in the document window on the right, as [Figure 4](#) shows. Note that this is the default set of initial components displayed, but you can change them to your own preferred components in the locations you choose. The Summary page lists the contents of an Object Explorer node or produces a report on the selected node. As you work more with the components, you'll invoke a third type of dialog, the non-modal dialog windows that are independent of Management Studio's main window.

Each of the component dialogs is dockable and hideable, just as in Visual Studio. A pushpin icon in the top button bar of each window provides an auto-hide option, which you can also reach by right-clicking at the top of the window. Automatically hiding windows can be useful when you need extra screen space for expanding the Object Browser, authoring queries, and so on. You can change the behavior of these dialogs by clicking the Window Position option, a small, triangular down arrow at the top right of each component dialog.

You can activate many other component dialogs from the View menu. The default Registered Servers and Object Explorer component dialogs are most useful for administration. Other component dialogs such as the Solution Explorer, Template Explorer, and Properties dialogs are more useful for query editing and analysis. Management Studio remembers your most recent windows settings and uses your last configuration when you restart the tool. If you want to reset your windows to the default setting, choose Reset Window layout from the Window menu.

Management Studio supports administration and script editing for multiple products: instances of the SQL Server database engine, along with instances of the Analysis Services server, Report Server, Integration Services, and SQL Server Mobile. You can register all these types of services in Management Studio's Register Servers component and manage them using Object Explorer. You can also edit T-SQL scripts, Analysis Services scripts, and SQL Server Mobile scripts and organize them in projects using the Solution Explorer component, as we show in the Query Editing section.

Administration

To fully appreciate the new features that Management Studio brings to DBAs, let's begin with a comparison of Management Studio with Enterprise Manager, the standard SQL Server 2000 administrative tool. Enterprise Manager is the essential SQL Server 2000 graphical utility for database server management, but it has several shortcomings. For example, it manages only the SQL Server relational engine and the SQL Agent service, but not Analysis Services. Further, Enterprise Manager can't display large numbers of servers, databases, or database objects efficiently. Another problem is that it often uses modal dialogs, so after starting some action such as a backup, you often have to bring up a second instance of Enterprise Manager to continue monitoring a server. Finally, Enterprise Manager is a Microsoft Management Console (MMC) application and isn't particularly stable, occasionally causing frustrating errors and unresponsive behavior.

Management Studio overcomes all these limitations. You can now manage Analysis Services instances in addition to the relational-engine instances. Management Studio uses the new SQL Management Objects (SMO) API for efficient management of large numbers of database objects. Finally, it uses non-modal dialogs whenever possible and is written in stable managed code.

Registered servers. Unlike with Enterprise Manager, the Management Studio component for registering servers and organizing those servers is separate from the component for exploring server objects. You can export a registered

server or server group's registration information to an XML configuration file, and then later import it into the same or another server, by just right-clicking over the registered server or server group and choosing Export.

SQL Server 2005 also separates SQL Server services management from server management. For example, you can't start or stop services from the Registered Servers component; to manage the services, you need to run the SQL Server Configuration Manager. Management Studio separates registering a server from connecting to a server and exploring the server's objects. Registering a server in the Registered Servers window makes a momentary connection for validation but doesn't keep you connected to the server. But if you right-click on a registered server and choose Object Explorer or New Query, SQL Server will use the connection information stored with the registered server for subsequent connections to connect to the server.

Object Explorer. Object Explorer is the main tool for organizing a server's objects. It provides both administrative and query-editing capabilities. As you drill down into a SQL Server 2005 database engine, for example, you'll notice that the resulting tree of options resembles Enterprise Manager. But you can also register and manage an Analysis Services instance, something not possible with the legacy Enterprise Manager.

From each of the nodes in the Object Explorer tree view, you can invoke various administrative dialogs. By right-clicking over a database server name, for example, you can invoke the non-modal Server Properties tabbed dialog, which lets you display and modify server administration features.

As you expand the Object Explorer tree on a SQL Server database engine instance, note that SQL Agent has its own node and that all T-SQL code objects fall under a Programmability node, which contains a node for assemblies. When you expand the tree on an Analysis Server, you'll also see a new node for stored procedures, which are also assemblies.

The Object Explorer dialogs are based on SMO, which replaces SQL Server 2000's SQL-DMO and provides a revised object model with added flexibility. SMO operates asynchronously, so opening server nodes with many objects or methods doesn't interfere with other activities. For example, you could be expanding a database node with hundreds or thousands of tables, and while it's expanding, you could open another node and perform some other work. Also, SMO contains an underlying scripting facility, which lets you script most actions you choose from Object Explorer.

Performing Database Administration Tasks

In Management Studio's Object Explorer, system objects are separate from user objects. For example, the system databases (master, model, msdb, and tempdb) are organized under System Databases, and the new AdventureWorks and AdventureWorksDW sample databases are user databases. Note that these sample databases don't install by default, and if you didn't choose the option to install them when setting up your SQL Server instance, you'll initially have no user databases available.

Similar to the dialog for server properties, you can set database options by invoking the Database Properties dialog. In addition, Management Studio supports the table and view designers (known as Visual Data Tools in SQL Server 2000), including the Database Diagrams tool, which Microsoft added in the April CTP. Management Studio dialogs that result in actions on database objects also now include options for scripting the action, as [Figure 5](#) shows.

Management Studio provides nodes in the Object Explorer for managing security. You can also manage legacy SQL Server 2000 DTS packages, Notification Services, database-level triggers, Database Mail, and server-level DDL triggers.

Management Studio provides the support for managing SQL Server Agent that you would expect, but also adds an important new node for managing SQL Agent proxies. A SQL Server Agent proxy determines the security context for an individual job step by providing SQL Agent the security credentials for a Windows user. T-SQL job steps don't normally use proxies, but you can manage job step proxies for ActiveX controls, xp_cmdshell operations, replication, Analysis Services, SQL Server, and SQL Server Integration Services jobs.

Query Editing

In addition to many new administrative features, Management Studio also gives you a query-editing environment that goes beyond the capabilities of SQL Server 2000's Query Analyzer.

For SQL Server 2000, Query Analyzer is the tool most appropriate for editing T-SQL code and scripts. Although Query Analyzer's T-SQL editor is excellent, it still has a few limitations. For example, it can't edit other kinds of SQL Server scripts, such as MDX queries. When you edit a query or script in Query Analyzer, the editing window must have a connection to a SQL Server, and you can't change an editing session's connection to another server. Query Analyzer's graphical showplan is a valuable tool for analyzing the performance of a query, but the graphical output isn't portable. Furthermore, you can edit T-SQL scripts from a disk file in Query Analyzer, but SQL Server 2000 has no interface

directly to version-control software. Finally, although you can use Query Analyzer's Object Browser to find T-SQL code objects in a database, you can't use Object Browser for any kind of database management.

Management Studio keeps almost all the editing capabilities of Query Analyzer, but also overcomes the limitations just mentioned. Management Studio can edit all types of SQL Server scripts and XML scripts, not just T-SQL. You can edit queries in a connected or disconnected mode, and you can change the connection over which a query is executed. You can transfer Management Studio's graphical showplan output for viewing in other instances of Management Studio, and read the graphical results without being connected to the original server. Management Studio's powerful Object Explorer has all the features of Query Analyzer's Object Explorer, with full support for administration.

Disconnected editing. You can start a new query in Management Studio in multiple ways. From the File, New menu options, you can create three types of new queries: a T-SQL query, an Analysis Services MDX, DMS, or XMLA query, or a SQL Server Mobile query. You can also start new queries from the Object Explorer, or from the Solution Explorer, which we'll cover next.

Query editing in Management Studio is connection-independent: You can start a query connected to a server, start disconnected and connect later, and you can change a connection for the same query. This makes editing queries from script and text files—already a best practice in SQL Server 2000—much more natural. Once you're in an editing mode, you should notice the additional query-editing icons, in particular those for connecting, disconnecting, and changing a connection. When you're editing a query in Management Studio, you might want to auto-hide the Object Browser window so that you have more room on the screen for editing.

Solutions and projects. Management Studio enhances the query-editing experience by letting you organize your script and text files into solutions and projects. A project is a named collection of script and text files that normally reside in a Windows file folder with the same name as the project. A solution is a collection of one or more projects, and you can include all the projects under a folder with the solution name or just have one project and one location in your solution. To start a new project, just navigate the Management Studio menu to File, New, Project, and you'll see the New Project dialog.

By default, Management Studio starts a solution with the same name as the project and places the solution and project in your My Documents folder tree. When you have many types of scripts to organize or a large project that needs dividing into many sub-projects, you can collect multiple projects into a distinctly named solution and specify your own location. You must create a project first, then specify a solution. (You can override the solution name's default setting.) Once you've created the new project and solution, you can navigate the solution and its projects by using the Solution Explorer component, as [Figure 6](#) shows.

For example, suppose you're planning to modify a number of database objects related to the AdventureWorks HumanResources schema. Some scripts will modify tables, some will modify views, and some will modify stored procedures. You could organize all these types of scripts in projects, and then all those projects in a solution. [Figure 6](#) shows one way of doing this: A solution called soln_HR contains projects for each type of HR schema change.

With solutions and projects, you can create collections of SQL Server Analysis Services scripts and SQL Mobile scripts, each in its own project. The collections you designate as projects will be created as folders on disk and will display as distinct nodes in Solution Explorer. You can assign to the project your own name and a unique disk location. By default, the project will be created in folders below the solution, and all files in a project will reside in the project's folder.

Organization is just one benefit of using solutions and projects for SQL Server script editing. Management Studio has a close integration with source-code control software, if the source-control system has a compatible plug-in. Visual SourceSafe (VSS) has a one-to-one mapping between Management Studio projects and VSS projects. After you create a Management Studio project and solution, you can check the entire solution into VSS and later check out individual files or projects. Just right-click over any node in the Solution Explorer, and the options for adding to a source-code control system will appear. You can customize source-control settings in the Options dialog from Management Studio's Tools menu. You can enable several other editing features, such as line numbers and dynamic help, from the same Tools, Options dialog.

Interactive database script editing. You can use Object Explorer to edit T-SQL objects in a database. When you drill down to database programmable objects such as stored procedures, functions, and triggers, you can choose either New or Modify to invoke the T-SQL query editor. The New option uses a template, which you can customize and browse with the Template Explorer component. The Modify option generates the appropriate ALTER script for the object you've chosen. The New and Modify options replace the Assisted Editors that existed in earlier beta versions of SQL Server 2005's Management Studio.

Showplan enhancements. Management Studio gives you two major enhancements to the viewing of query plans. You can view the graphical version of a showplan just as in Query Analyzer, but in Management Studio the icons and their

colors have been revised. In addition, most icons show many details of what SQL Server is doing as it performs the operation the icon indicates. You can zoom in on a large query plan by using the plus (+) sign in the lower right corner of the graphical query plan window. Also, as you select any given node in a query plan, the Properties dialog displays additional information about the step.

You'll find a great new feature of Management Studio in the ability now to make your graphical showplans portable. You start by directing output to XML using SET SHOWPLAN_XML ON. If you execute the query to a grid, the results set will contain a link to the XML document containing the showplan information. You can view the showplan output in Management Studio's XML editor by clicking on the link.

You can also export the XML to a portable format for viewing graphically. Just right-click on the XML link in the result set and save the output to a file with a ".sqlplan" extension, as [Figure 7](#) shows. If you open the saved file from the File, Open menu option, Management Studio displays the XML document as a graphical query plan, as [Figure 8](#) shows.

Note that the Properties dialog continues to display additional information from the query plan. With this capability, you can email a query plan to a colleague or support provider—and she or he can view it without having to connect to your database.

Bringing It Together

The tools available with SQL Server 2005 will enhance your management and development experience. In particular, the SQL Server Management Studio is customizable so that once you decide which features are useful for you, you can hide the features that you don't need. Although you might encounter a brief-but-steep learning curve when first encountering the new tools, it will be well worth your while to spend some time familiarizing yourself with their vast list of possibilities.

Figure 1

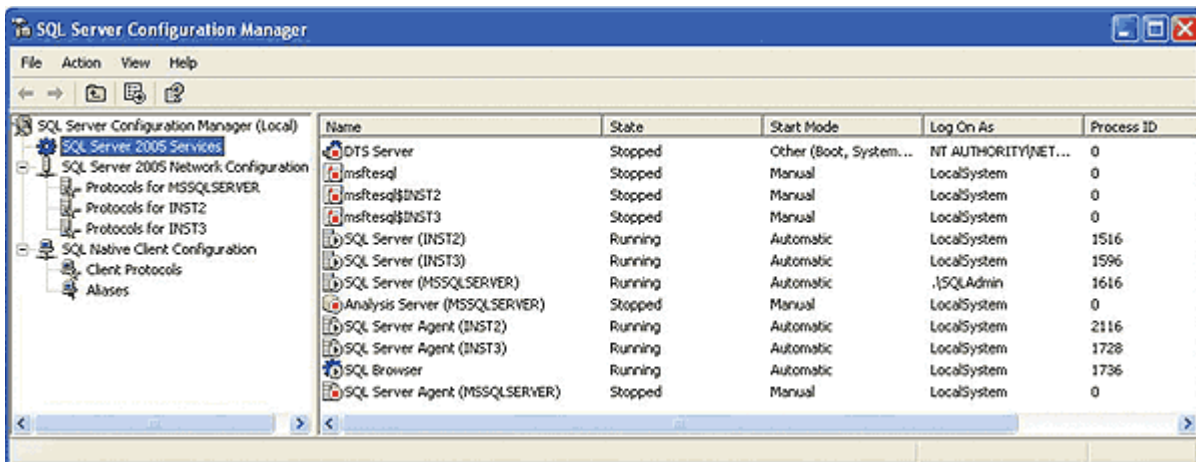


Figure 2

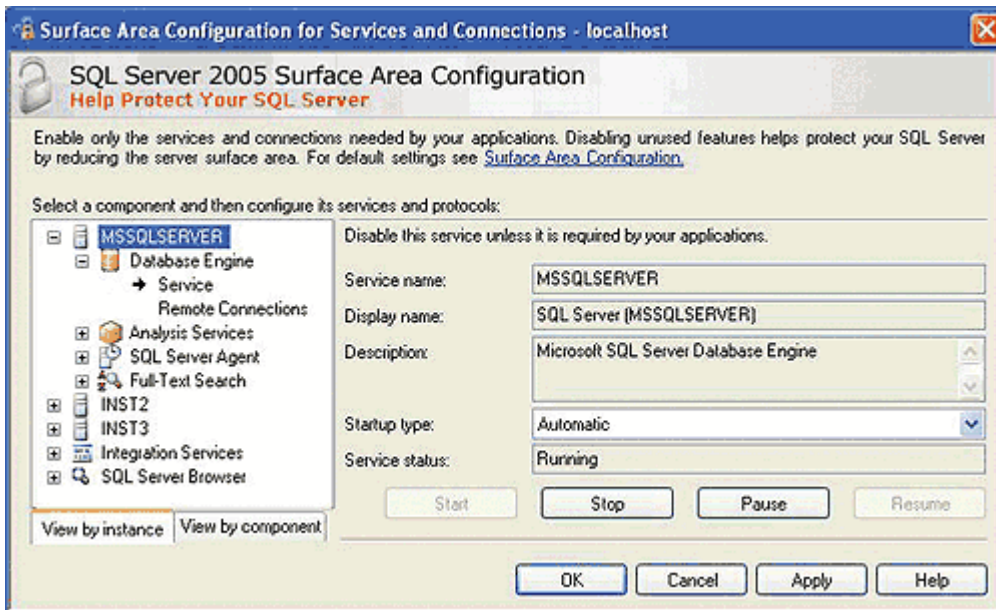


Figure 3

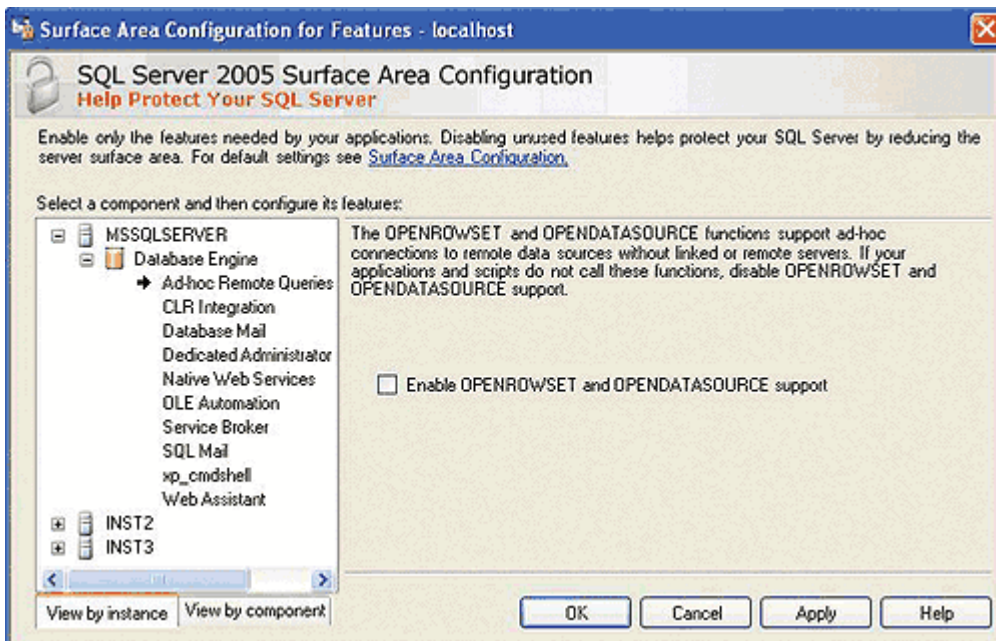


Figure 4

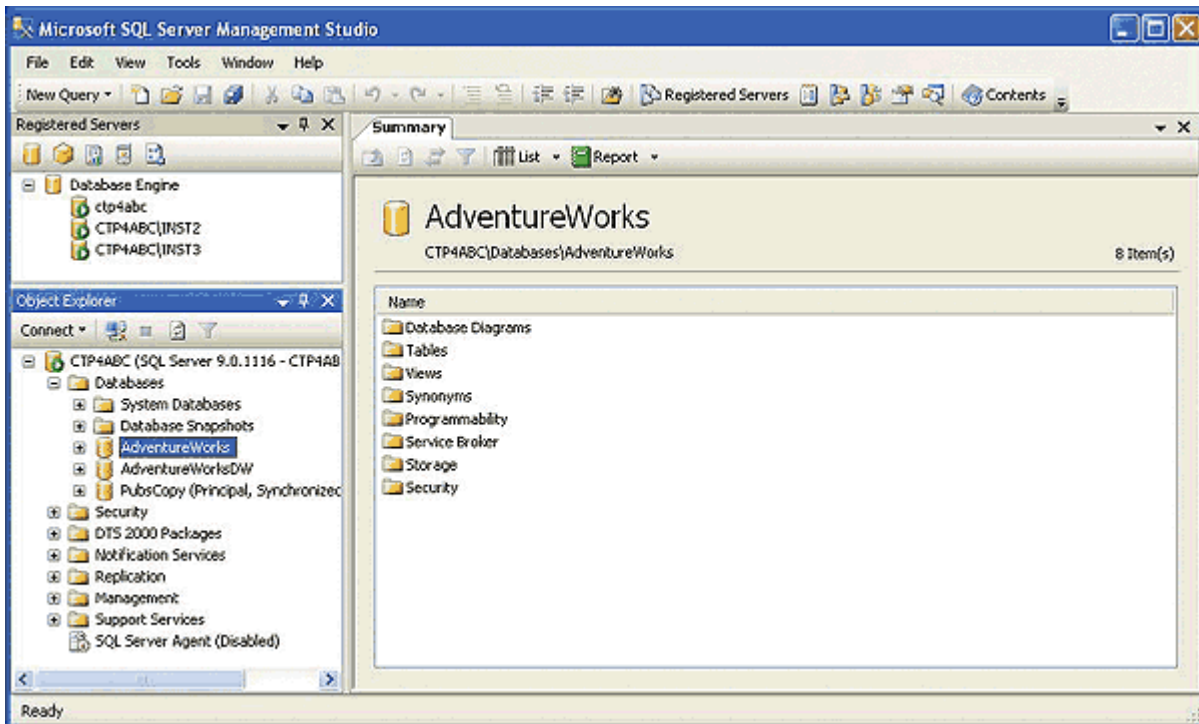


Figure 5

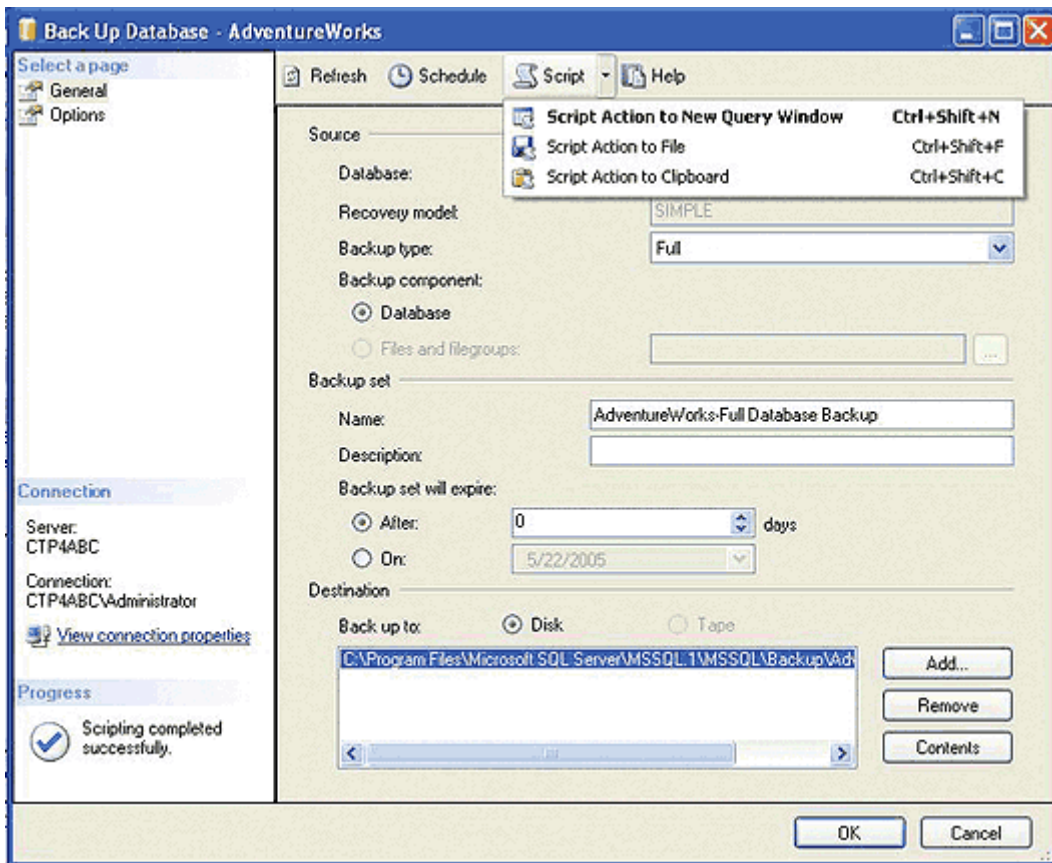


Figure 6

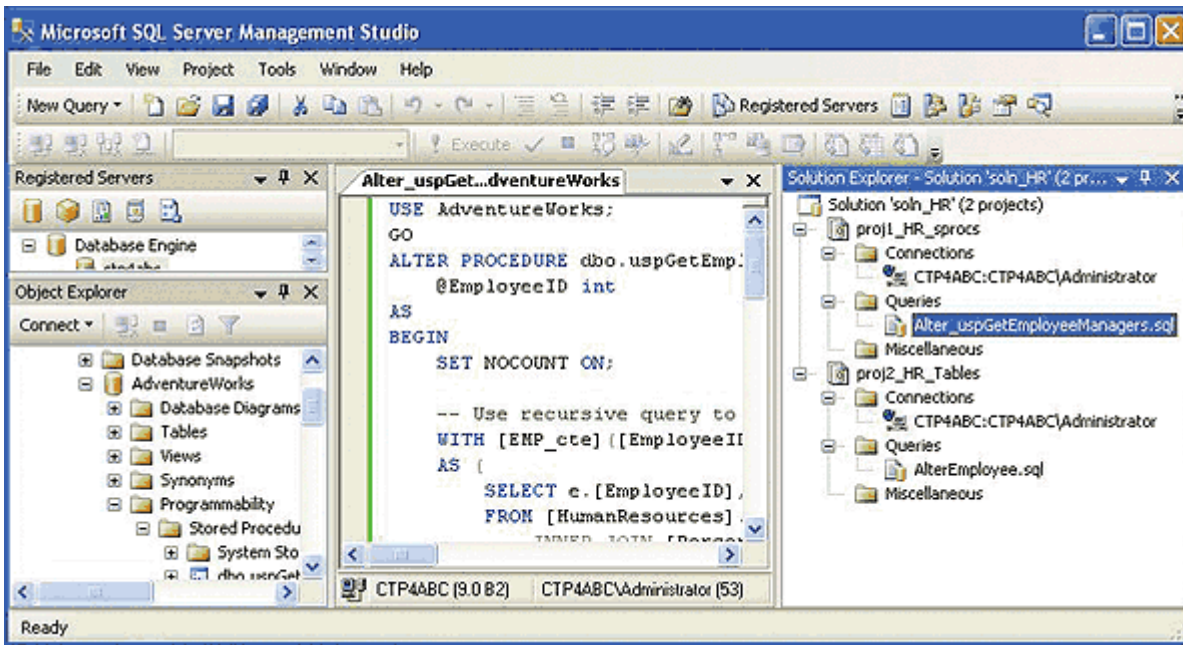


Figure 7

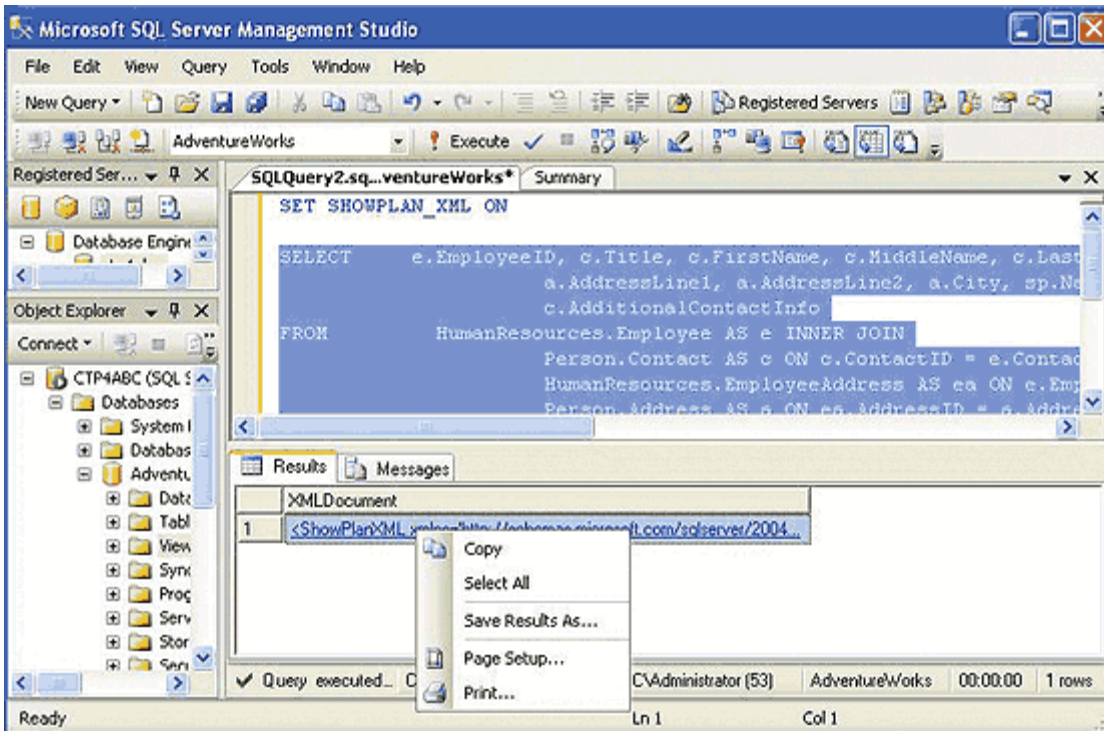


Figure 8

