CHAPTER 21

Monitoring SQL Server 2005

Monitoring SQL Server 2005 is key to ensuring the system stays up and operational with as few unplanned interruptions as possible. When problems do occur, monitoring ensures that problems are quickly identified and corrected. Problems left unattended can quickly grow into major issues if not dealt with.

With the built-in SQL Server 2005 native monitoring tools, you can configure alerts for conditions and examine some of the historical information that has been logged on the server. With an external solution such as Microsoft System Center Operations Manager (OpsMgr) 2007, monitoring can be taken to the next level by leveraging built-in knowledge for hundreds of common problems, performance can be tracked and reported, and uptime reports can be easily generated.

Monitoring SQL Server with Native Tools

SQL Server 2005 provides several built-in tools that assist in your ongoing monitoring efforts. Database administrators commonly use these tools to verify the different SQL Server components are running correctly and to troubleshoot problems as they are encountered. SQL Server 2005 also introduces a significant improvement in the way notifications are sent so you can be alerted when specific events occur. The following sections provide a look into several monitoring tools and demonstrate how to set up each tool.

Monitoring Job Activity

The Job Activity Monitor allows the monitoring of all agent jobs for a specific SQL Server instance through the SQL Server

Management Studio (SSMS). To view all jobs with the Job Activity Monitor, follow these steps:

1. From the test server (SQL01), choose Start, All Programs, Microsoft SQL Server 2005, SQL Server Management Studio.
2. Select Database Engine from the Server Type drop-down; then enter the server and instance name (**`SQL01\INSTANCE01`**).
3. Select Windows Authentication from the Authentication drop-down menu and then click the Connect button.
4. A connection to the Database Engine is made. If the Object Explorer pane is not visible, press the F8 button.
5. Expand the SQL Server Agent container.
6. Right-click Job Activity Monitor.
7. Select View Job Activity.

Within the Job Activity Monitor, each job hosted by the SQL Server instance is listed. The columns above the display fields can be used to sort the different jobs. Both the Filter link located in the status pane and the Filter button located at the top of the window can be used to filter the list of agent jobs.

Filter settings can be applied to each of the agent job columns. This capability is helpful when many jobs are listed. To apply a filter to the list of jobs, follow these steps:

1. From within the Job Activity Monitor, click the Filter button or the View Filter Settings link.
2. To configure the filter to show only failed jobs, select Failed from the Last Run Outcome drop-down.
3. When the filter is configured, enable the Apply Filter option near the bottom of the window.
4. Figure 21.1 shows how the filter settings should look when configured. Click OK to accept the settings.

Note

The filter icon changes from blue to green when a filter is applied to the list. To remove the filter, simply disable the Apply Filter option from within the Filter Settings dialog box.

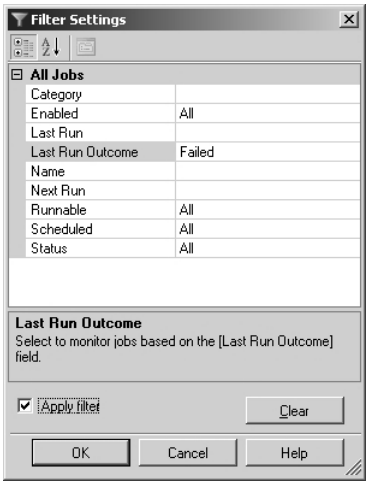

FIGURE 21.1
Job Activity Monitor filter settings.

The details window does not update automatically; however, you can configure it by selecting View Refresh Settings from the status pane. Note that the refresh interval and the filter settings are not persistent. When the Job Activity Monitor is closed, the settings revert back to the defaults.

The jobs shown in the details pane can also be managed. The right-click context menu allows you to start, stop, enable, disable, delete, and view the job history. You also can access the properties of the job by right-clicking the job and selecting Properties.

Configuring Database Mail

Mail delivery in SQL Server 2005 has significantly changed from previous versions. Although the legacy SQL Mail functionality is still available, it has been deprecated and should not be used.

Database Mail replaces SQL Mail and has been improved in almost every aspect. For example, Database Mail no longer requires an installation of a MAPI client such as Outlook on the server just to send email; email is now sent using standard Simple Mail Transfer Protocol (SMTP). This also means one or more available SMTP servers in the organization can be used to relay mail, which could include an existing Exchange environment.

To use the new Database Mail feature, the user must be part of the DatabaseMailUserRole role in the MSDB database. This role allows the execution of the `sp_send_dbmail` stored procedure.

Installing an SMTP Server

This demonstration leverages a locally installed SMTP server on the test server SQL01. Follow these steps to install the server:

1. Select Start, Control Panel, Add or Remove Programs.
2. Click Add/Remove Windows Components.
3. Select Application Server and then click Details.
4. Select Internet Information Services (IIS) and then click Details.
5. Select SMTP Service, as shown in Figure 21.2, and then click OK.
6. Click OK and then click Next to start the install.
7. Insert the Windows disk if prompted and then click OK.
8. Click Finish when the installation process is complete.

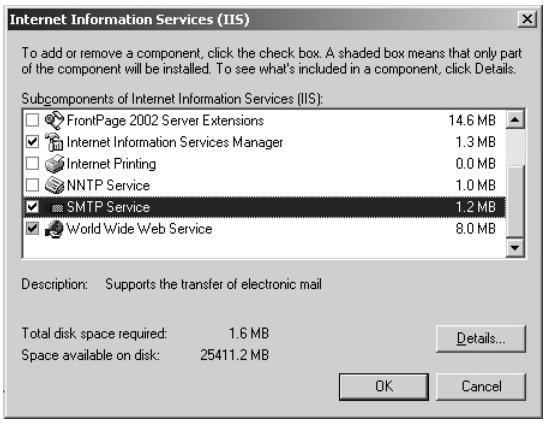

FIGURE 21.2
SMTP component of IIS.

After installing the SMTP server, you must configure it. The configuration allows SQL Server to connect and send mail. Follow these steps to configure the security on the SMTP server:

1. Select Start, Administrative Tools, Internet Information Services (IIS) Manager.

2. Expand SQL01 (local computer).
3. Right-click Default SMTP Virtual Server.
4. Select Properties.
5. Select the Access tab and then click the Relay button.
6. Click the Add button.
7. Type the IP address of server SQL01 and then click OK. You can find the IP address of the server by selecting Start, Run; then type **`CMD`** in the Open field and click OK. Type **`IPCONFIG`** in the command window, and the output will show the IP Address of the server.
8. Click OK and then close the console.

This configuration allows SQL Server to send mail through the SMTP server without any restrictions on the destination. Remember, if the IP address of the server changes, the delivery of mail will break.

> **Note**
>
> Do not select the All Except the List Below option in the relay configuration. This option, which opens the SMTP server so that anyone can relay mail, is considered a security risk.

Implementing Database Mail

After installing and configuring the SMTP server, follow these steps to configure Database Mail for INSTANCE01 on server SQL01:

1. From within SSMS, expand Management. Database Mail should be listed.
2. Right-click Database Mail and select Configure Database Mail.
3. On the Welcome page, click Next.
4. Select the Set Up Database option and click Next.
5. If prompted, click Yes to enable Database Mail.
6. Type **`Email Notification`** in the Profile Name field.

> **Note**
>
> The previous steps allow you to enable Database Mail functionality for the SQL Server 2005 instance. The Database Mail functionality can also be enabled and disabled through the SQL Server Surface Area Configuration tool.

The next step is to establish a Database Mail account, which is simply a list of SMTP servers used to send the email.

Multiple Database Mail accounts can be used. When email is sent, each mail account is tried in order until one of them is successful. When the email is successfully sent through an account, that account is used for subsequent email delivery until it becomes unavailable.

Each account can be configured with a different authentication, depending on the requirements of the environment. The SMTP server installed on SQL01 allows only the local server to send mail without authentication. No other server is able to relay off this system. Follow these steps to add the Database Mail account:

1. Click the Add button to open the New Database Mail Account page.
2. Type **`Local SMTP`** in the Account Name field.
3. Type **`dbSupport@companyabc.com`** for the email address.
4. Type **`Email Notification`** in the Display Name field.
5. The Server Name field must be populated with the name of your SMTP server. Type **`SQL01`** in the Server Name field.
6. Click OK and then click Next.

Figure 21.3 shows how the New Database Mail Account page should look. You can add additional accounts using the same procedure.

On the Manage Profile Security page, you can configure the profile as public or private. Public profiles can be used by any user in the DatabaseMailUserRole role, whereas private profiles can be used only by specific database users. The profile can also be configured as the default profile.

To continue using the wizard, enable the Email Notification Profile by checking the Public check box and then click Next.

On the Configure System Parameters page, you can configure the setting that controls how Database Mail operates. For example, to configure the system to retry delivery if an error is experienced, set the Account Retry Attempts and Account Retry Delay (Seconds) options.

To continue the wizard, accept the default values and click Next. Click Finish to complete the wizard and execute the defined configuration.

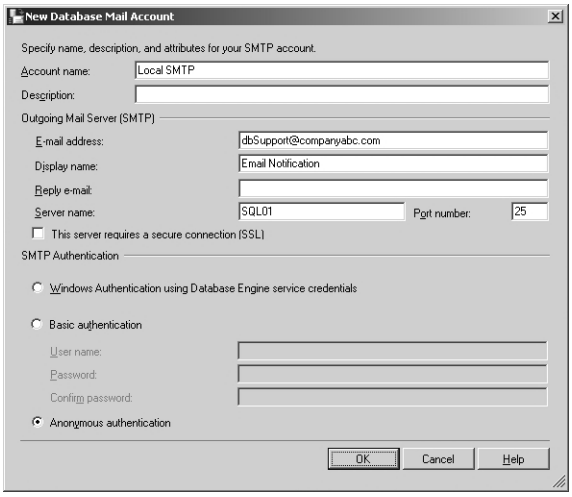

FIGURE 21.3
New Database Mail account.

Validating the Database Mail Configuration

To test the email delivery and validate email is working correctly, follow these steps from within SSMS:

1. Right-click Database Mail.
2. Select Send Test E-Mail.
3. Select Email Notification as the profile.
4. Enter an email address in the To field.
5. Click Send Test E-mail and then click OK.

The Database Mail log can be used to validate that the email was sent from the SQL Server to the SMTP server. To view the log, right-click the Database Mail container and select View Database Mail Log from the menu.

The following stored procedures can be used to configure Database Mail through the data definition language (DDL):

- `sysmail_add_account_sp`
- `sysmail_add_profile_sp`
- `sysmail_add_profileaccount_sp`

- sysmail_add_principalprofile_sp
- sp_send_dbmail

For example, you can now use the following code to send an email notification:

```
EXEC msdb.dbo.sp_send_dbmail @recipients='user@companyabc.com',
    @profile_name = 'Email Notification',
    @subject = 'Test Email Notification',
    @body = 'Email message from SQl01\INSTANCE01',
    @body_format = 'HTML';
```

Adding Operators

An operator is a user or a group that receives notifications. Notifications can include email, pagers, and net send. The schedule of the operator can also be configured; for example, an operator can be defined to receive notification during business hours, and a different operator can be defined to use notifications during nonbusiness hours or on the weekend.

From within SSMS, you can define new operators. To add a new operator to the SQL Server instance, follow these steps:

1. From within SQL Server Management Studio, expand SQL Server Agent.
2. Right-click Operators and select New Operator.
3. Enter the name of the operator in the field provided.
4. Enter the email address in the Email Name field.
5. Enable a suitable schedule for the operator. For example, enable from 8am to 9pm Monday through Friday.
6. Click OK.

You can use the Notification section of the New Operator page to enable notifications for existing alerts on the server.

Defining Alerts

Alerts can be defined for a wide range of SQL Server events. You can receive alerts on the following types of events:

- SQL Server events
- SQL Server performance conditions
- WMI events

For example, when log shipping is implemented, alerts of the type SQL Server Event Alert are defined and the error number being responded to is 14420. Follow these steps to generate an alert when the used log file space falls below 100MB in the AdventureWorks database:

1. From within SQL Server Management Studio, expand SQL Server Agent.
2. Right-click Alerts and select New Alert.
3. In the Name field, type **`AW Log Files Used Size`**.
4. Select SQL Server Performance Condition Alert from the Type drop-down.
5. Select MSSQL$INSTANCE01:Databases for the object. The first part of the object corresponds to the instance name; if the SQL Server was installed as the default instance, the object name is SQLServer:Databases.
6. Select Log File(s) Used Size (KB) for the counter.
7. Select AdventureWorks for the instance.
8. Select Falls Below for the alert condition.
9. Enter **`102400`** for the value.

You can also define a response to an alert. Responses can include executing a job or notifying an operator. The following steps demonstrate how to add an operator to the previously created alert:

1. From within the New Alert window, select the Response option page.
2. Enable the Notify Operators option.
3. Enable the Email column for the operator created earlier.
4. Click OK to finish creating the alert.

You can use the Options page of the new alert to specify whether the error text is included in the different types of alerts.

Using the SQL Server Profiler

The SQL Server Profiler tool captures SQL Server 2005 events as they are generated on a SQL Server. The captured information, referred to as a *workload*, can be reviewed in the UI or saved to a trace file. The workload can be used to analyze performance or can be replayed to conduct N+1 testing. The SQL Server Profiler tool is invaluable for getting detailed insight into the

internal workings of applications and databases from a real-world and real-time perspective.

For additional information on using the SQL Server Profiler, see Chapter 22, "Performance Tuning and Troubleshooting SQL Server 2005" (online).

Using the Database Engine Tuning Advisor

The Database Engine Tuning Advisor automates the process of selecting an optimized set of indexes, indexed views, statistics, and partitions and even provides the code to implement the recommendations it makes. The Database Engine Tuning Advisor can work with a specific query or can use a real-world workload as gathered by the SQL Server Profiler tool. The advantage of the latter approach is that the workload is generated based on actual usage, and the tuning recommendations reflect that.

The Database Engine Tuning Advisor is customizable and allows you to select the level of recommendation that the tool recommends. This feature allows you to maintain the existing database design and make appropriate fine-tuning recommendations for just indexes. Or you can make the existing design flexible and then have the tool recommend far-reaching changes to the structure such as partitioning.

For additional information on using the Database Engine Tuning Advisor, see Chapter 22.

Monitoring SQL Logs

SQL Server 2005 keeps several different logs detailing the various processes that take place on the server. All the log files can be viewed through the Log File Viewer.

The SQL Server error logs are the primary logs kept for instances. By default, six archive logs and one active log are kept. A new log file is created each time an instance is started.

Follow these steps to access the SQL Server logs through SSMS:

1. From the test server (SQL01), choose Start, All Programs, Microsoft SQL Server 2005, SQL Server Management Studio.
2. Select Database Engine from the Server Type drop-down; then enter the server and instance name (**`SQL01\INSTANCE01`**).

3. Select Windows Authentication from the Authentication drop-down menu and then click the Connect button.
4. A connection to the Database Engine is made. If the Object Explorer pane is not visible, press the F8 button.
5. From within the Object Explorer pane, expand Management, SQL Server Logs.

You can change the number of SQL error logs kept by right-clicking the SQL Server Logs container in the Object Explorer and selecting Configure. From within the Configure SQL Server Error Logs window, enable the option to limit the number of error log files and specify the number of error log files.

The SQL Server 2005 Agent error logs keep track of agent processes that take place on the SQL Server. If a problem with a SQL Server Agent process occurs, these logs should be checked to help determine the cause of the issue.

Nine Agent archive logs and one current log are kept. To access the Agent error logs from within the SSMS, expand the SQL Server Agent container and then expand the Error Logs container. You can configure the Agent error logging levels by right-clicking on the Error Logs container and selecting Configure. By default, only error and warning messages are enabled. To enable informational logging messages, simply select the check box and click OK.

Note

Enabling informational logging may significantly increase the size of the log files.

By right-clicking either an Agent or SQL Server error log and selecting View Log, you can open the Log File Viewer. The Log File Viewer allows you to view each log file individually. A powerful feature of the Log File Viewer is to combine the log files, including the Windows event log, into a single view. You can accomplish this by enabling and disabling the logs from the menu pane on the left side of the window. Figure 21.4 shows the current SQL Server logs combined with the current Windows logs.

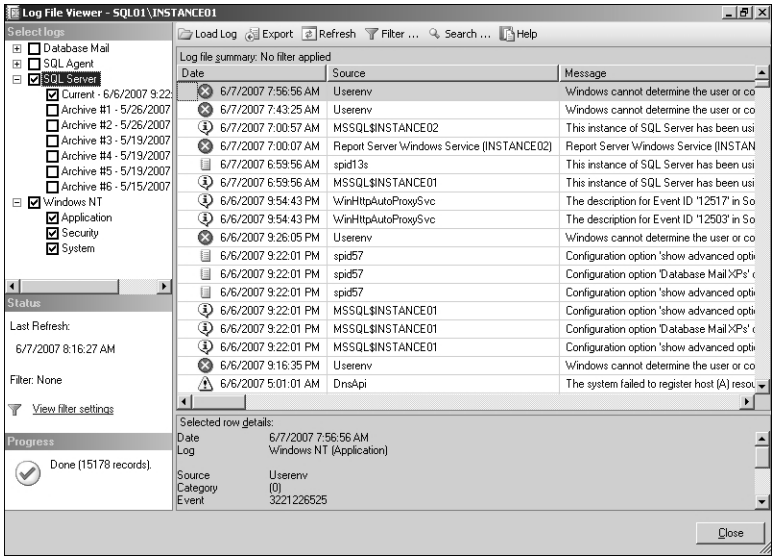

FIGURE 21.4
Log File Viewer.

Using OpsMgr 2007 to Proactively Monitor SQL Server 2005

System Center Operations Manager (OpsMgr) 2007 provides an excellent approach to monitoring and managing SQL Server 2005. OpsMgr helps to identify problems before they evolve into critical issues through the use of OpsMgr's event monitoring, performance monitoring, and alerting features.

OpsMgr provides a real-time view, shown in Figure 21.5, of critical SQL events and intelligently links them to appropriate Microsoft Knowledgebase articles. Cryptic event IDs are directly matched to known issues and immediately referred to technical reference articles in Microsoft's Knowledgebase for troubleshooting and problem resolution. OpsMgr monitors SQL Servers and other Windows-based servers and applications using standard Windows services such as Windows Management Instrumentation (WMI) and Windows logged events. In addition, OpsMgr also provides a reporting feature that allows network administrators to track problems and trends occurring on their network. Reports can be generated automatically, providing network administrators a quick, real-time view of their server performance data.

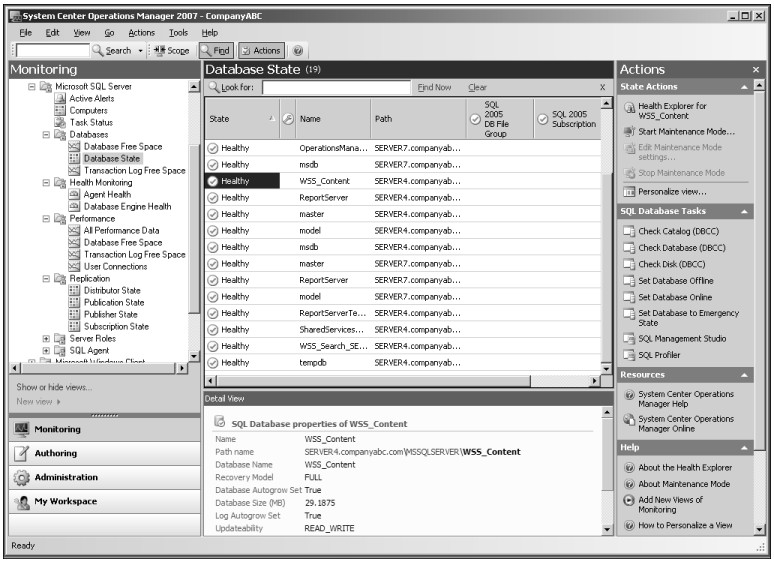

FIGURE 21.5
Monitoring SQL Servers using the OpsMgr 2007 console.

Note

System Center Operations Manager was originally developed by NetIQ and then purchased and released as Microsoft Operations Manager (MOM) 2000. OpsMgr was subsequently updated and released as MOM 2005. Recently, the product has been completely redesigned and was released as System Center Operations Manager 2007. OpsMgr 2007 contains powerful management capabilities and presents a fundamental change in the way systems are monitored. In addition to individual server monitoring, groups of systems can now be monitored together as a service with multiple interdependent and distributed components.

OpsMgr integrates with and manages Windows Server 2003 and SQL Server 2000 and 2005. It can also be used in Windows 2000 Server or mixed environments to provide for automated monitoring of vital network functionality. This type of functionality is instrumental in reducing downtime and getting the most out of a SQL Server 2005 investment. In a nutshell, OpsMgr is an effective way to gain proactive, rather than reactive, control over a mixed Windows Server 2003/Windows 2000 environment.

The following sections focus on defining OpsMgr as a monitoring component for SQL Server. These sections provide specific analysis of the way OpsMgr operates and presents OpsMgr design best practices, specific to deployment for SQL Server monitoring. In addition, they describe how to install and configure the SQL 2005 Management Pack into a System Center Management group.

Explaining How OpsMgr Works

OpsMgr is an event and performance data–driven monitoring system that effectively allows for large-scale management of mission-critical servers. Organizations with a medium to large investment in SQL Servers or other Windows servers will find that OpsMgr allows for an unprecedented ability to keep on top of the tens of thousands of event log messages that occur on a daily basis. In its simplest form, OpsMgr performs two functions: processing gathered events and performance data, and issuing alerts and automatic responses based on those data.

OpsMgr provides for several major pieces of functionality as follows:

- **Event Log Monitoring**—OpsMgr Agents, deployed on managed systems, monitor for specific event log information. This data is used for reporting, auditing, and monitoring of the specific events.
- **Advanced Alerting Capabilities**—OpsMgr provides advanced alerting functionality by enabling email alerts, paging, and functional alerting roles to be defined.
- **Performance Monitoring**—OpsMgr collects performance statistics that can let an administrator know whether a server is being overloaded or is close to running out of disk space, among other things.
- **Built-In Application-Specific Intelligence**—OpsMgr management packs are packages of information about a particular application or service, such as SQL 2000/2005, Windows Server 2003, FRS, DNS, DHCP, Exchange Server, or other applications. The Microsoft management packs are written by the design teams for each individual product, and they are loaded with the intelligence and information necessary to properly troubleshoot and identify problems.

Processing Operational Data

OpsMgr manages Windows Server 2003 networks through monitoring rules used for Windows event monitoring, object discovery, performance data gathering, and application specific synthetic transactions. Monitoring rules define how OpsMgr collects, handles, and responds to the information gathered.

OpsMgr monitoring rules handle incoming event data and allow OpsMgr to react automatically, either to respond to a predetermined problem scenario, such as a failed hard drive, with a predefined action (trigger an alert, execute a command or script) or to consolidate multiple events into one event that correlates a group of related events. The monitoring rules also enable OpsMgr to automatically determine which events are important to a network administrator, minimizing administrative overhead.

Generating Alerts and Responses

OpsMgr monitoring rules can generate alerts based on critical events, synthetic transactions, or performance thresholds that are met or exceeded. An alert can be generated by a single event or by a combination of events or performance thresholds. For example, a failed fan can generate a simple alert, whereas a failed database generates a more complex alert relating to the failed SQL Server database services and IIS web pages that rely on the availability of the database. Alerts can also be configured to trigger responses such as email, pages, Simple Network Management Protocol (SNMP) traps, and scripts to notify you of potential problems.

OpsMgr can be configured to notify various IT groups. For example, an alert triggered by a failed database can alert the help desk with email and you with email and a paged message. In brief, OpsMgr is completely customizable in this respect and can be modified to fit most alert requirements.

Outlining OpsMgr Architecture

OpsMgr is primarily composed of four basic components: the operations database, reporting database, management server, and OpsMgr agents. OpsMgr was specifically designed to be scalable and can subsequently be configured to meet the needs of any size company. This flexibility stems from the fact that all OpsMgr components can either reside on one server or can be distributed across multiple servers.

Each of these various components provides specific OpsMgr functionality. OpsMgr design scenarios often involve the separation of parts of these components onto multiple servers. For example, the database components can be delegated to a dedicated server, and the management server can reside on a second server.

Understanding How OpsMgr Stores Captured Data

OpsMgr itself utilizes two Microsoft SQL Server databases for all collected data. This data includes event log and performance data gathered from each

managed computer. The operations database also stores all the monitoring rules and scripts used by the management pack. This database must be installed as a separate component from OpsMgr but can physically reside on the same server, if needed. The reporting database stores data for long-term trend analysis and can potentially grow much larger than the operations database. Proper SQL procedures and maintenance for the database components are critical for proper OpsMgr functionality.

Determining the Role of Agents in System Monitoring

The *agents* are the monitoring components installed on each managed computer. They use the collected event logs and performance counter data and process them based on the management pack rules installed on the computer.

Creating Administrative Boundaries with Management Groups

OpsMgr utilizes the concept of *management groups* to logically separate geographical and organizational boundaries. Management groups allow you to scale the size of OpsMgr architecture or politically organize the administration of OpsMgr. Each management group consists of the following components:

- One operations database
- One reporting database
- One or more management servers
- Managed agents

OpsMgr can be scaled to meet the needs of different sized organizations. For small organizations, all the OpsMgr components can be installed on one server with a single management group.

In large organizations, on the other hand, the distribution of OpsMgr components to separate servers allows the organizations to customize their OpsMgr architecture. Multiple management groups provide load balancing and fault tolerance within the OpsMgr infrastructure. Organizations can set up multiple management servers at strategic locations, to distribute the workload between them.

Note

The general rule of thumb with management groups is to start with a single management group and add on more management groups only if they are absolutely necessary. Administrative overhead is reduced, and there is less need to re-create rules and perform other redundant tasks with fewer management groups.

How to Use OpsMgr

Using OpsMgr is relatively straightforward. It can be configured through three sets of consoles: an operations console, a web console, and a command shell. The operations console provides full monitoring of agent systems and administration of the OpsMgr environment while the web console provides access only to the monitoring functionality. The command shell is written on PowerShell and provides command-line access to administer the OpsMgr environment. After a OpsMgr environment is deployed, very little needs to be done to monitor the system; it is easy to forget that OpsMgr is deployed. The real value of the system presents itself when a critical system event is logged and you are notified.

Managing and Monitoring with OpsMgr

As mentioned in the preceding section, two methods can be used to configure and view OpsMgr settings. The first approach is through the operations console and the second is through the command shell. With the Administration section of the operations console, you can easily navigate through a hierarchical tree structure and configure the security roles, notifications, and configuration settings. With the Monitoring section of the operations console, quick "up/down" status and overall environment health can be monitored easily.

In addition to the operations console, a Web-based administration console with a web browser such as Microsoft Internet Explorer (versions 4.01 or higher) can be used to view monitoring information. Through the web console, you can review the status of managed systems with the ability to take action on and update alerts. Access to the associated Knowledgebase is provided as well.

Reporting from OpsMgr

OpsMgr has a variety of preconfigured reports and charts. The reports allow you a quick review of the status of systems and services on the network.

They can also help you monitor your networks based on performance data. The reports can be run on demand or at scheduled times. OpsMgr can also generate HTML-based reports that can be published to a web server and viewed from any web browser. Vendors can also create additional reports as part of their management packs.

Using Performance Monitoring

Another key feature of OpsMgr is the capability to monitor and track server performance. OpsMgr can be configured to monitor key performance thresholds through rules that are set to collect predefined performance data, such as memory and CPU usage over time. Rules can be configured to trigger alerts and actions when specified performance thresholds have been met or exceeded, allowing network administrators to act on potential hardware issues. Performance data can be viewed from the OpsMgr operator's console. Performance threshold rules can also be configured to watch performance counters to establish a baseline for the environment and then alert when the counter subsequently falls outside the defined baseline.

Exploring the SQL Server Management Pack

When imported, the SQL Server management pack automatically discovers the following objects on managed servers in the management group:

- SQL Server 2005 DB Engine
- SQL Server 2000 DB Engine
- SQL Server 2005 Analysis Services
- SQL Server 2005 Reporting Services
- SQL Server 2005 Integration Services
- SQL Server 2005 Distributor
- SQL Server 2005 Publisher
- SQL Server 2005 Subscriber
- SQL Server 2005 DB
- SQL Server 2000 DB
- SQL Server 2005 Agent
- SQL Server 2000 Agent
- SQL Server 2005 Agent Jobs
- SQL Server 2000 Agent Jobs

- SQL Server 2005 DB File Group
- SQL Server 2005 DB File

As you can see OpsMgr finds many of the components associated with a SQL Server and not just the server itself. Availability statistics of each component can be calculated independently or together as a group. For example, an availability report can be scheduled for a single database on a server or the entire server. This type of discovery also allows each component to be placed into maintenance mode independently of other components on the server. For example, a single database can be placed into maintenance mode to prevent alerts from being generated while the database is worked on or repaired while other databases on the server are still being monitored.

In addition to basic monitoring of SQL Server related events and performance data, the SQL Server management pack provides advanced monitoring through custom scripts associated with rules in the management pack. The following rules are specific to SQL Server monitoring. Each rule can be customized for the environment or even a specific server being monitored.

- **Block Analysis**—When an SPID is blocked for more than one minute, an alert is generated. This detection can be configured through the Blocking SPIDs monitor associated with the SQL 2005 DB Engine object.
- **Database Configuration**—SQL Server specific configurable options such as Auto Close, Auto Create Statistics, Auto Shrink, Auto Update, DB Chaining, and Torn Page Detection. This detection can be configured through the corresponding configuration monitors associated with the SQL 2005 DB object.
- **Database Health**—Tracks the availability and current state of databases on SQL Servers in the environment. This detection can be configured through the Database Status monitor associated with the SQL Server 2005 DB object.
- **Database and Disk Space**—The free space within database and transaction logs are monitored. An alert is event generated when predefined thresholds are exceeded or a significant change in size is detected. This detection can be configured through the corresponding performance monitors associated with the SQL Server2005 DB object.
- **Long-running Agent Jobs**—Agent jobs that run for more than 60 minutes will generate an alert by default. This detection can be

configured through the Long Running Jobs performance monitor associated with the SQL 2005 Agent object.

- **Service Pack Compliance**—The current service pack level can be monitored by configuring the Service Pack Compliance configuration monitor associated with the SQL Server 2005 DB Engine object. An alert is generated when a server is not at the required service pack level.

Within the Monitoring area of the Operators console the following views are available to assist with monitoring the environment:

- Alerts View
- Computers View
- Database Free Space Performance
- Transaction Log Free Space Performance
- Database State
- Agent Health State
- Database Engine Health State
- Analysis Services State
- Database Engines State
- Integration Services State
- Reporting Services State
- SQL Agent Job State
- SQL Agent State

The SQL Server management pack also includes several default reports to help with trend-specific SQL:

- SQL Broker Performance
- SQL Server Database Counters
- SQL Server Configuration
- SQL Server Lock Analysis
- SQL Server Servicepack
- SQL User Activity
- Top 5 Deadlocked Databases
- User Connections by Day

- User Connections by Peak Hours
- SQL Database Space Report

The latest version of management packs should always be used because it includes many improvements and updates from the release code.

Integrating OpsMgr with Legacy Management Software

Network management is not a new concept. Simple management of various network nodes has been handled for quite some time through the use of the SNMP. Quite often, simple or even complex systems that utilize SNMP to provide for system monitoring are in place in an organization to provide for varying degrees of system management on a network.

OpsMgr can be configured to integrate with these network systems and management infrastructures. Special connectors can be created to provide bidirectional information flows to other management products. OpsMgr can monitor SNMP traps from SNMP-supported devices as well as generate SNMP traps to be delivered to third-party network management infrastructures. In addition, OpsMgr can also monitor live events on Unix systems using the syslog protocol.

Understanding OpsMgr Component Requirements

Each OpsMgr component has specific design requirements, and a good knowledge of these factors is required before beginning the design of a OpsMgr. Hardware and software requirements must be taken into account, as well as factors involving specific OpsMgr components such as service accounts and backup requirements.

Exploring Hardware Requirements

Having the proper hardware for OpsMgr to operate on is a critical component of OpsMgr functionality, reliability, and overall performance. Nothing is worse than overloading a brand-new server only a few short months after its implementation. The industry standard generally holds that any production servers deployed should remain relevant for three to four years following deployment. Stretching beyond this time frame may be possible, but the ugly truth is that hardware investments are typically short term and need to be replaced often to ensure relevance. Buying a less-expensive server may save

money in the short term but could potentially increase costs associated with downtime, troubleshooting, and administration. That said, the following are the Microsoft-recommended minimums for any server running OpsMgr 2007:

- 1.8Ghz+ Pentium or compatible processor
- 20GB of free disk space
- 2GB of random access memory (RAM)

These recommendations apply only to the smallest OpsMgr deployments and should be seen as minimum levels for OpsMgr hardware. Future expansion and relevance of hardware should be taken into account when sizing servers for OpsMgr deployment.

Determining Software Requirements

OpsMgr can be installed on Windows Server 2003 SP1 or R2 editions. The database for OpsMgr must be run on a Microsoft SQL Server 2005 database. The database can be installed on the same server as OpsMgr or on a separate server, a concept that is discussed in more detail in following sections.

OpsMgr itself must be installed on a member server in a Windows Server 2003 (or Windows 2000) Active Directory domain, and *not* on a domain controller, because it does not physically install if this is the case. It is most often recommended to keep the installation of OpsMgr on a separate server or set of separate dedicated member servers that do not run any other separate applications.

A few other factors critical to the success of a OpsMgr implementation are as follows:

- DNS must be installed in the environment to utilize mutual authentication.
- SQL 2005 Reporting Services or higher must be installed for an organization to be able to produce custom reports using OpsMgr's reporting feature.

OpsMgr Backup Considerations

Like most technical implementations, OpsMgr includes several key components that require regular backups for disaster recovery scenarios. The system state and system drive of each OpsMgr server should be backed up to provide for quick recovery of OpsMgr configuration information. Special add-ons to backup software specifically written for OpsMgr can ensure the ability to

back up live data from OpsMgr systems. At this time, many of the large backup software manufacturers offer this type of specialized add-on to their products, and it would be prudent to integrate these components into a OpsMgr design.

In addition, the most critical piece of OpsMgr, the SQL database, should be regularly backed up using an additional add-on to standard backup software that can effectively perform online backups of SQL databases. If integrating these specialized backup utilities into a OpsMgr deployment is not possible, it becomes necessary to periodically dismount the OpsMgr database and perform offline backups. Either way, the importance of backups in a OpsMgr environment cannot be overstressed.

Deploying OpsMgr Agents

OpsMgr agents are deployed to all managed servers through the OpsMgr configuration process. These agents can be configured to be automatically installed for all Windows Servers on a specific domain based on managed computer rules. These rules use the Fully Qualified Domain Name (FQDN) of the computer and the domain to allow you to select which systems should have the client installed automatically. You can use wildcards to specify a broad range of computers. Certain situations, such as monitoring across firewalls, can require the manual installation of these components.

Advanced OpsMgr Concepts

OpsMgr's simple installation and relative ease of use often betray the potential complexity of its underlying components. This complexity can be managed, however, with the right amount of knowledge of some of the advanced concepts of OpsMgr design and implementation.

Dedicated Management Server Versus All-in-One Server

As previously mentioned, OpsMgr components can be divided across multiple servers to distribute load and ensure balanced functionality. This separation allows OpsMgr servers to come in four potential "flavors," depending on the OpsMgr components held by those servers. The four OpsMgr server types are as follows:

- **Operations Database Server**—An operations database server is simply a member server with SQL Server 2005 installed for the OpsMgr operations database. No other OpsMgr components are installed on this server. The SQL Server 2005 component can be

installed with default options and with the System Account used for authentication. Data in this database is kept for four days by default.

- **Reporting Database Server**—A reporting database server is simply a member server with SQL Server 2005 and SQL Server Reporting Services installed. This database stores data collected through the monitoring rules for a much longer period than the operations database and is used for reporting and trend analysis. This database requires significantly more drive space than the operations database server. Data in this database is kept for 13 months by default.
- **Management Server**—A management server is the communication point for both management consoles and agents. Effectively, a management server does not have a database and is often used in large OpsMgr implementations that have a dedicated database server. Often, in these configurations, multiple management servers are used in a single management group to provide for scalability and to address multiple managed nodes.
- **All-in-One Server**—An all-in-one server is effectively an OpsMgr server that holds all OpsMgr roles, including that of the databases. Subsequently, single-server OpsMgr configurations use one server for all OpsMgr operations.

Multiple Management Groups

As previously defined, an OpsMgr management group is a logical grouping of monitored servers that are managed by a single OpsMgr SQL database, one or more management servers, and a unique management group name. Each management group established operates completely separately from other management groups, although they can be configured in a hierarchal structure with a top-level management group able to see "connected" lower-level management groups.

The concept of connected management groups allows OpsMgr to scale beyond artificial boundaries and also gives a great deal of flexibility when combining OpsMgr environments. However, certain caveats must be taken into account. Because each management group is an island in itself, each must subsequently be manually configured with individual settings. In environments with a large number of customized rules, for example, such manual configuration would create a great deal of redundant work in the creation, administration, and troubleshooting of multiple management groups.

Deploying Geographic-Based Management Groups

Based on the factors outlined in the preceding section, it is preferable to deploy OpsMgr in a single management group. However, in some situations it is preferable *not* to divide an OpsMgr environment into multiple management groups, or dividing it this way is unavoidable.

The most common reason for division of OpsMgr management groups is division along geographic lines. In situations in which WAN links are saturated or unreliable, it may be wise to separate large "islands" of WAN connectivity into separate management groups.

Simply being separated across slow WAN links is not enough reason to warrant a separate management group, however. For example, small sites with few servers would not warrant the creation of a separate OpsMgr management group, with the associated hardware, software, and administrative costs. However, if many servers exist in a distributed, generally well-connected geographical area, that may be a case for the creation of a management group. For example, an organization could be divided into several sites across the United States but decide to divide the OpsMgr environment into separate management groups for East coast and West coast, to roughly approximate their WAN infrastructure.

Smaller sites that are not well connected but are not large enough to warrant their own management group should have their event monitoring throttled to avoid being sent across the WAN during peak usage times. The downside to this approach, however, is that the reaction time to critical event response is increased.

Deploying Political or Security-Based Management Groups

The less common method of dividing OpsMgr management groups is by political or security lines. For example, it may become necessary to separate financial servers into a separate management group to maintain the security of the finance environment and allow for a separate set of administrators.

Politically, if administration is not centralized within an organization, management groups can be established to separate OpsMgr management into separate spheres of control. This would keep each OpsMgr management zone under separate security models.

As previously mentioned, a single management group is the most efficient OpsMgr environment and provides for the least amount of redundant setup, administration, and troubleshooting work. Consequently, artificial OpsMgr division along political or security lines should be avoided, if possible.

Sizing the OpsMgr Database

All new technologies seem to consume tremendous amounts of disk space, and OpsMgr is no exception to this trend. Depending on several factors, such as the type of data collected, the length of time that collected data will be kept, or the amount of database grooming that is scheduled, an OpsMgr database can grow by leaps and bounds, if left unchecked. An organization might want, for example, to keep event information for longer periods of time, which would drive up the size of the database exponentially.

It is important to monitor the size of the database to ensure that it does not increase well beyond the bounds of acceptable size. The following actions can be taken to reduce the size of an OpsMgr database:

- **Archive collected data**—The more often old data is archived, the smaller a database will become, for obvious reasons. When an OpsMgr database becomes too large, for example, it may become necessary to archive old data to alternate storage mediums. The downside to this approach, however, is the fact that reporting can generate historical reports only up to the point of the last archival. Finding the right trade-off between an aggressive archiving schedule and an expansive database is recommended.
- **Modify the grooming interval**—As evident from the formula presented in the sidebar, increasing the database grooming interval decreases the size of a database significantly. Setting the grooming interval to once every few days, for example, can aggressively address space limitations and keep the database consistent. Setting a regular grooming interval is subsequently key to an effective database maintenance strategy.

OpsMgr can be configured to monitor itself, supplying advance notice of database problems and capacity thresholds. This type of strategy is highly recommended because OpsMgr could easily collect event information faster than it could get rid of it.

Defining Capacity Limits

As with any system, OpsMgr includes some hard limits that should be taken into account before deployment begins. Surpassing these limits could be cause for the creation of new management groups and should subsequently be included in a design plan. These limits are as follows:

- **Single Operations Database Per Management group**—OpsMgr operates through a principle of centralized, rather than distributed, collection of data. All event logs, performance counters, and alerts are sent to a single centralized database, and there can subsequently be only a single operations database per management group. Considering the use of a backup and high-availability strategy for the OpsMgr database is therefore highly recommended, to protect it from outage.
- **Management Servers** —OpsMgr does not have a hard-coded limit of management servers per management group. However it is recommended to keep the environment between three to five management servers.
- **2000 Agents Per Management Server**—Each management server can theoretically support up to 2000 monitored agents for every OpsMgr server. In most configurations, however, it is wise to limit the number of agents per management server, although the levels can be scaled upward with more robust hardware, if necessary.
- **50 Active Console Instances**—OpsMgr does not limit the number of instances of the Web and Operator consoles; however, going beyond the suggested limit may introduce performance and scalability problems.

Defining System Redundancy

In addition to the scalability built into OpsMgr, redundancy is built into the components of the environment. Proper knowledge of how to deploy OpsMgr redundancy and place OpsMgr components correctly is important to the understanding of OpsMgr redundancy.

Having multiple management servers deployed across a management group allows an environment to achieve a certain level of redundancy. If a single management server experiences downtime, another management server within the management group will take over the responsibilities for the monitored servers in the environment. For this reason, it may be wise to include multiple management servers in an environment to achieve a certain level of redundancy if high uptime is a priority.

Because there can be only a single OpsMgr database per management group, the database is subsequently a single point of failure and should be protected from downtime. Utilizing Windows Server 2003 clustering or third-party fault-tolerance solutions for SQL databases helps to mitigate the risk involved with the OpsMgr database.

Securing OpsMgr

Security has evolved into a primary concern that can no longer be taken for granted. The inherent security in Windows Server 2003 is only as good as the services that have access to it; therefore, it is wise to perform a security audit of all systems that access information from servers. This concept holds true for management systems as well because they collect sensitive information from every server in an enterprise. This includes potentially sensitive event logs that could be used to compromise a system. Consequently, securing the OpsMgr infrastructure should not be taken lightly.

Physically Securing OpsMgr

Aside from actual software security, one of the most important forms of security is actual physical security. OpsMgr servers should be physically secured behind locked doors, and login access to the console should be curtailed to help protect the critical information contained within the environment. This concept cannot be overstressed because physical security is one of the most highly overlooked but yet one of the most critical components of a secure infrastructure.

In addition to physical security, OpsMgr servers should be carefully locked down at the OS level to prevent unauthorized access. This includes the creation of complex passwords for service accounts and the application of the latest service packs and security updates using the automatic update features in Windows Server 2003 to help keep the environment secure and up to date. In addition, administration of OpsMgr security can be greatly simplified via the creation of an Active Directory group that controls OpsMgr administration. This group can be granted admin rights to OpsMgr servers, and users can be added as members to this group. Simplifying the administration of security often strengthens security as well because administrators take fewer security shortcuts when troubleshooting problems.

Securing OpsMgr Agents

Each server that contains an OpsMgr agent and forwards events to OpsMgr servers has specific security requirements. Server-level security should be established and should include provisions for OpsMgr data collection. All traffic between OpsMgr components, such as the agents, management servers, and database, are encrypted automatically for security, so the traffic is inherently secured.

In addition, environments with high security requirements should investigate the use of encryption technologies such as IPSec to scramble the event IDs

that are sent between agents and OpsMgr servers, to protect against eavesdropping of OpsMgr packets.

Basic Firewall Requirements

OpsMgr servers that are deployed across a firewall have special considerations that must be taken into account. Port 5723, the default port for OpsMgr communications, must specifically be opened on a firewall to allow OpsMgr to communicate across it. In addition, OpsMgr servers can be specifically configured to exist in a DMZ firewall configuration, as long as the proper access is granted to the managed servers from the DMZ.

Outlining Service Account Security

In addition to the aforementioned security measures, security of an OpsMgr environment can be strengthened by the addition of multiple service accounts to handle the different OpsMgr components. For example, the Action account and the SDK account should be configured to use separate service accounts, to provide for an extra layer of protection in the event that one account is compromised.

Downloading and Extracting the SQL 2005 Management Pack for OpsMgr 2007

As previously mentioned, management packs contain intelligence about specific applications and services and include troubleshooting information specific to those services. This management pack is required for effective monitoring of a SQL Server environment using OpsMgr 2007.

To install the SQL 2005 Management Pack on an OpsMgr management server, first download it from the Microsoft downloads page at www.microsoft.com/technet/prodtechnol/mom/catalog/catalog.aspx?vs=2007.

To install each management pack on the OpsMgr management server, follow these steps:

1. Double-click on the downloaded executable.
2. Select I Agree to the license agreement and click Next to continue.
3. Select a location to which to extract the management pack and then click Next.
4. Click Next again to start the installation.
5. Click Close when the file extraction is complete.

Importing the Management Pack File into OpsMgr 2007

After extracting the management pack, follow these steps to upload the management pack files directly into the OpsMgr administrator console:

1. From the OpsMgr Console, navigate to the Administration node.
2. Click the Import Management Packs link.
3. From the Select Management Packs to Import dialog box, browse to the location where the files were extracted and select all of them. Click Open.
4. From the Import Management Packs dialog box, shown in Figure 21.6, click Import.
5. Click Close when finished.

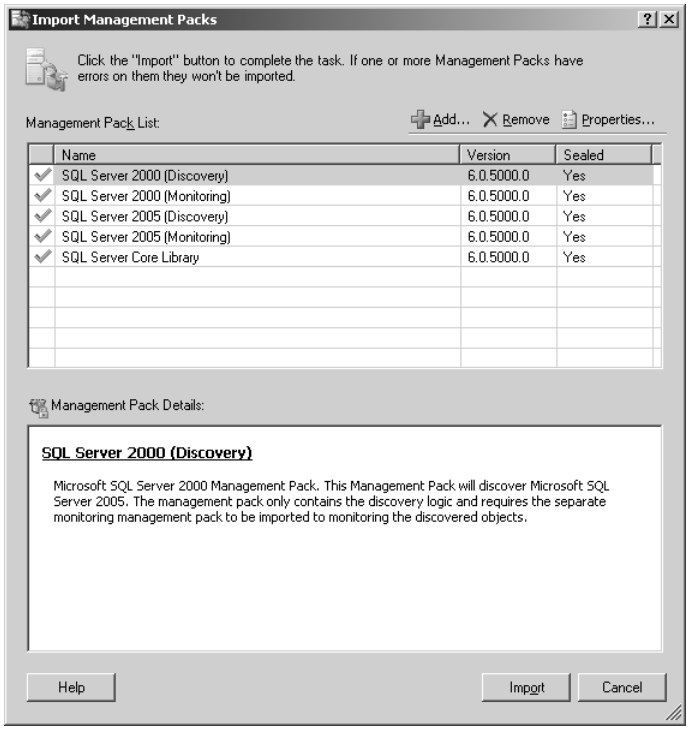

FIGURE 21.6
Beginning the SQL management pack import process.

Installing the OpsMgr Agent on the SQL Server

Installation of OpsMgr agents on SQL Server can be automated from the OpsMgr console. To initiate the process of installing agents, follow these steps:

1. From the OpsMgr 2007 Console, click the Monitoring node.
2. Click the Required: Configure Computers and Devices to Manage link.
3. From the Computer and Device Management Wizard, shown in Figure 21.7, select Next to start the process of deploying agents.

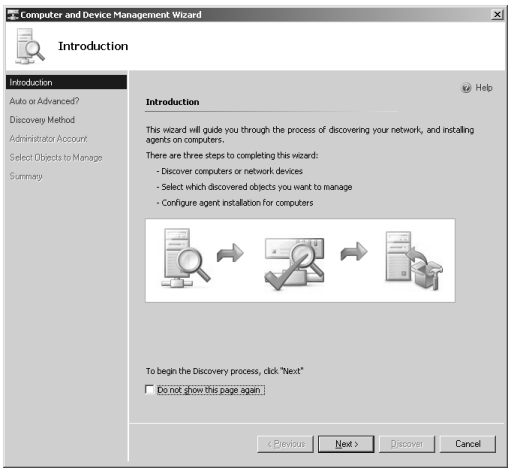

FIGURE 21.7
Deploying agents to SQL Servers.

4. On the Auto or Advanced dialog box, select Automatic Computer Discovery or experiment by doing a selective search. Note that Automatic Computer Discovery can take awhile and have a network impact. Click Next to continue.
5. Enter a service account to perform the search; it must have local admin rights on the boxes where the agents will be installed. You can also select to use the Action account. Click Discover to continue.
6. After Discovery, a list of discovered servers is displayed, as shown in Figure 21.8. Check the boxes next to the servers where the agents will be installed and click Next.

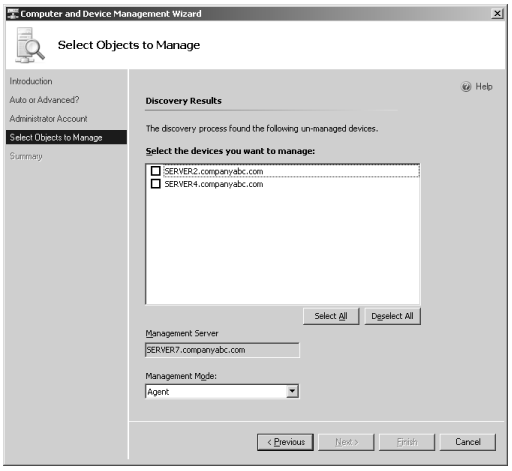

FIGURE 21.8
Selecting servers to deploy the agents to.

7. On the summary page, leave the defaults and click Finish.
8. Click Close when complete.

After completing the installation, you might need to wait a few minutes before the information from the agents is sent to the console.

Monitoring SQL Functionality and Performance with OpsMgr

After the management pack is installed for SQL and the agent has been installed and is communicating, OpsMgr consolidates and reacts to every event and performance counter sent to it from the SQL Server. This information is reflected in the OpsMgr operations console, as shown in Figure 21.9.

For more information on OpsMgr 2007, see the Microsoft website at www.microsoft.com/opsmgr.

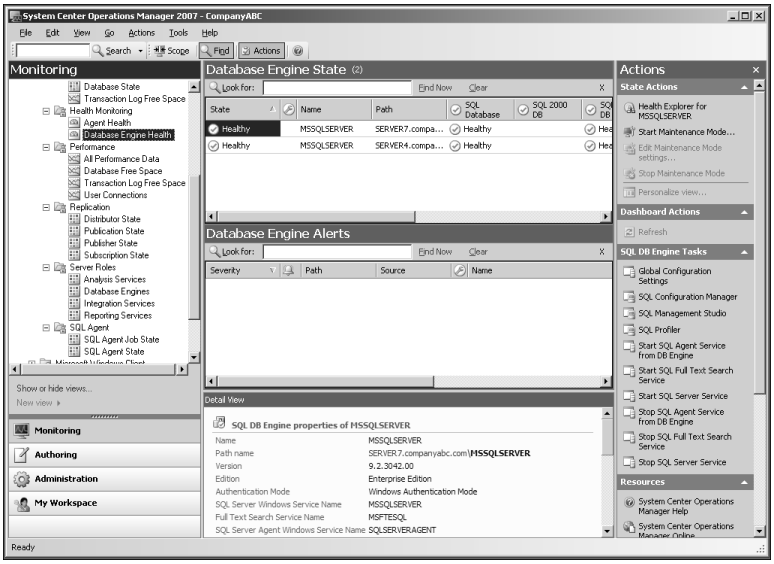

FIGURE 21.9
Monitoring SQL functionality in the OpsMgr 2007 console.

Summary

The built-in monitoring tools provide a limited amount of proactive monitoring by allowing you to configure events as necessary to alert operators. Built-in monitoring tools also provide a historical analysis through logs, greatly assisting the troubleshooting process.

System Center Operations Manager 2007 is an ideal monitoring and management platform for a SQL farm and has proven its value in proactively identifying potential server issues before they degrade into server downtime. OpsMgr for SQL provides the built-in reliability of the OS and allows for greater control over a large, distributed server environment. In addition, proper understanding of OpsMgr components, their logical design and configuration, and other OpsMgr placement issues can help an organization to fully realize the advantages that OpsMgr can bring to a SQL Server 2005 environment.

Best Practices

- When a centralized monitoring solution is unavailable, leverage the built-in Database Mail functionality to generate notifications for important conditions that may arise in the environment.
- Use the SQL Server Profiler to generate workloads and test performance of databases.
- Use the Database Engine Tuning Advisor to assist in the creation of indexes, indexed views, statistics, and partitions for the database and to test the results of the changes.
- When automated monitoring is unavailable be sure to keep a close watch on SQL Server logs as important information about the state of the environment is detailed.
- Examine the use of System Center Operations Manager 2007 for monitoring SQL Servers.
- Install the updated SQL 2005 Management Pack into the OpsMgr management group.
- Take future expansion and relevance of hardware into account when sizing servers for OpsMgr deployment.
- Keep the installation of OpsMgr on a separate server or set of separate dedicated member servers that do not run any other separate applications.
- Use SQL Server Reporting Services to produce custom reports using OpsMgr's reporting feature.
- Start with a single management group and add on additional management groups only if they are absolutely necessary.
- Use a dedicated service account for OpsMgr.
- Use a database volume of at least 5GB depending on the length of time needed to store events.
- Monitor the size of the OpsMgr database to ensure that it does not increase beyond the bounds of acceptable size.
- Archive collected data.
- Modify the grooming interval to aggressively address space limitations and keep the database consistent.
- Configure OpsMgr to monitor itself.