CHAPTER 3

How Does It Work?

IN THIS CHAPTER

Successfully deploying Microsoft Operations Manager (MOM) 2005 requires understanding how it works and how to implement it. MOM 2005 is a powerful tool, but that power comes at the expense of a certain amount of complexity. In this chapter, we continue our introduction to MOM 2005 with an architectural overview and discussion of some of its major components. We discuss several MOM components including agents, the MOM Service, and the Data Access Service.

You'll become familiar with terms such as management group, management server, managed computer, and management packs. If you have already read Chapter 2, "What's New," you may notice that some of these components were previously introduced as new functionality with MOM 2005. This chapter provides the groundwork for understanding MOM, which will assist in planning your installation and deployment of MOM.

Architectural Overview

MOM obtains raw event and performance data to translate into system health information. Using rules, which are MOM's basic unit of instruction, you can define the characteristics of a properly running application and have MOM warn you when these capabilities are not being met. MOM collects event and performance data on monitored systems, looking for specific events that indicate poor performance, errors, or other factors specified in its rules.

MOM not only collects data, it filters that information so that you see only what is important. MOM also consolidates multiple occurrences of events into a single representation—minimizing superfluous "noise" and data. Alerts occur when specific events or performance conditions occur.

Monitoring begins after installing one or several management groups, importing management packs, enabling rules, and identifying computers to monitor. The monitored computers

send event and performance data to a management server, which stores that data in the MOM database. Operational data is viewed using the MOM Operator console or a web-based console. Data can also be maintained in a reporting database for long-term analysis and study.

What Is a Management Group?

The basic management unit of MOM is the MOM *management group,* illustrated in Figure 3.1. A management group is a MOM installation that includes one MOM database, one or more MOM management servers, and MOM agents installed on monitored systems. You can optionally install a Report server, Reporting database server, and/or additional management console(s). MOM can also manage a limited number of computers using an agentless monitoring technique.

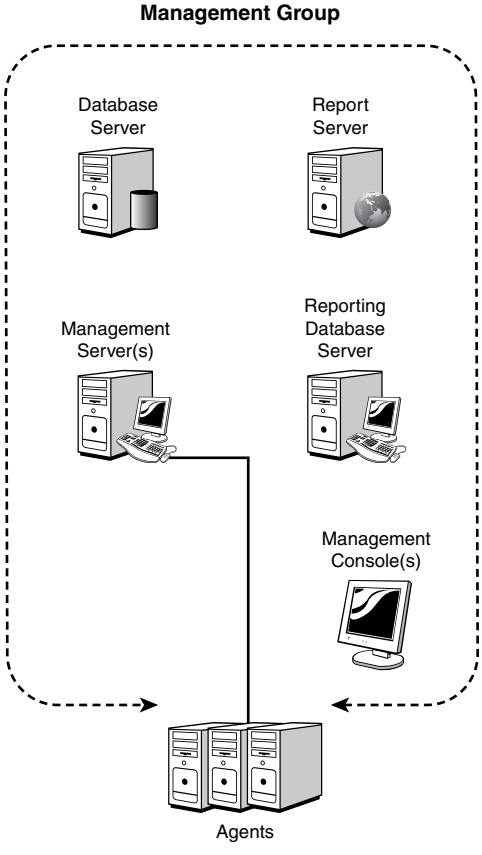

FIGURE 3.1 The MOM management group.

Management Group Names

A management group is identified by a unique alphanumeric name, which is specified when you install a management server. The management group name cannot be

modified after the management server is built; changing the name requires removing all MOM components and reinstalling them. The group name is up to you, although it is often advantageous to base them on geographical locations, organizational departments, or administrative needs.

The management group provides the features and benefits discussed in Chapter 1, "Operations Management Basics," namely:

- Event-based monitoring
- Easily deployed and scalable infrastructure
- Effective system availability and performance tracking

There are quite a few MOM components, all of which are ultimately bound into a management group. A functional management group contains the following components:

- Operations database
- Management Server
- Data Access Service (DAS)
- Agents
- Administrator console
- Operator console

There are optional components, including:

- Reporting database
- Reporting console
- Web console

In the next section we look at these components and how they interoperate.

Server Roles

The MOM components can be grouped into server roles (shown in Figure 3.2), which is ultimately what is built during your MOM implementation. The standard component groupings are

- **Database server role**—The database server normally contains the operations database and is a platform optimized for data collection—that is, to rapidly process a large amount of incoming data from the management servers. In classic client server architecture, this is known as the *backend tier*. In the MOM architecture, the database server is a big portion of the data layer or tier.

- **Management server role**—The management server typically contains the MOM service, the DAS, an agent, the Administrator console, the Operations console, and the Web console. The management server handles most of the centralized business logic and presentation layer functions, with the important exception of the reporting functions.
- **Report server role**—The report server typically contains the reporting database and the Reporting console. The reporting database is frequently located on a separate server for performance reasons. Using a separate server allows large volumes of data to be retained and mined with the reporting function, without affecting the operation's function on the database and management servers.

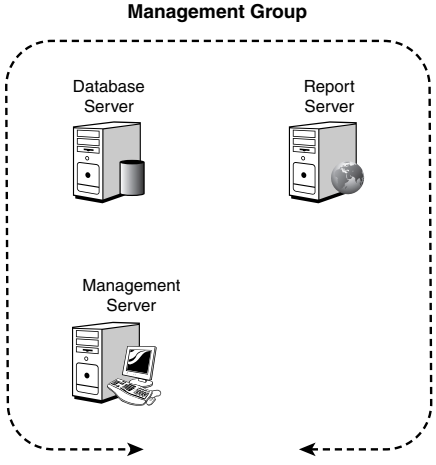

FIGURE 3.2 Management group roles.

In any given MOM installation, you can combine these roles in a variety of ways. In smaller installations, the server roles might all be on the same physical server with the net effect that there is only one "MOM Server." In a medium-sized organization, there might be two physical management servers for fault tolerance and a single database server with both the operations and reporting functions. In a large enterprise organization, there would likely be separate physical servers for the database server and the reporting database server, as well as multiple management servers for both load balancing and fault tolerance (and possibly also clustering the databases). Chapter 4, "Planning Your MOM Deployment," covers the rationale for splitting or combining the roles, as well as how to create a MOM 2005 design that meets the needs of your organization.

A management group can have multiple management servers as shown in Figure 3.3. Reasons for having multiple management servers include:

- Scalability
- Fault tolerance
- Security
- Crossing geographic or network boundaries

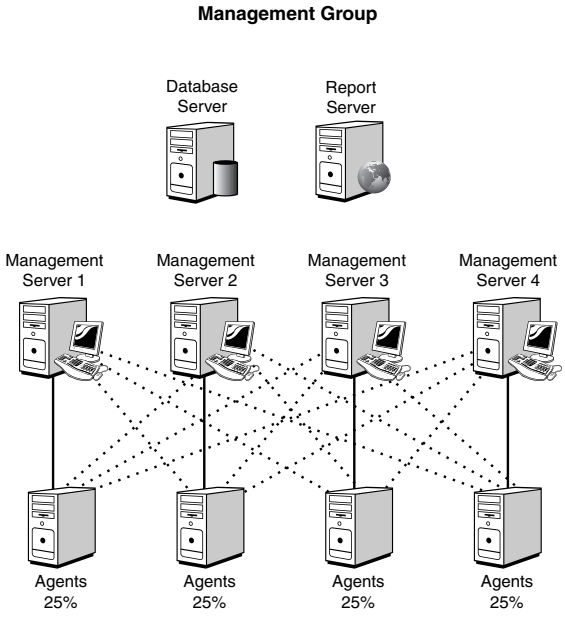

FIGURE 3.3 Management group with four management servers.

Agents within the management group have a primary management server assigned to them, and then each of the remaining management servers are backup management servers for the agent. Figure 3.3 shows four management servers with each assigned 25% of the agents, represented by the solid lines. The mesh of dotted lines represents the redundant failover connections the agents are automatically configured to use. Agent configurations are dynamically adjusted as new management servers are brought online or decommissioned. In the event that an agent fails over to one of its backup management servers, it periodically checks to see whether its primary management server is back online and fails back automatically when that occurs. More information on failover processing is available in Appendix A, "MOM Internals."

Communications

As you can see in Figure 3.4, MOM uses a variety of communications methods that are optimized for security and efficiency. Notice that the communications between the management server and the agent are different depending on the direction of the communication. This has important ramifications for firewall support and security, which we will discuss later in this section.

For the Remote Procedure Calls (RPC)/Distributed Component Object Model (DCOM) protocols, RPC uses Transmission Control Protocol (TCP) port 135, and DCOM uses a nightmarish combination of TCP, User Data Protocol (UDP), ports, and connections.

DCOM is particularly troublesome for firewall access because it dynamically assigns ports to processes. By default, it freely assigns TCP and UDP ports ranging from 1024 to 65535, making it difficult to function securely across a firewall. In addition, new connections are established when responding to a client, meaning that the port the client used for the request is not the same as the port used for the response. Also, DCOM does not support Network Address Translation (NAT), which is among the more common methods of configuring a firewall. You can configure DCOM to only use TCP, restrict the ports the client and server use, and open up the firewall just enough to get the communications through. However, the bottom line is these actions seriously compromise the security of your firewall and the communications across it.

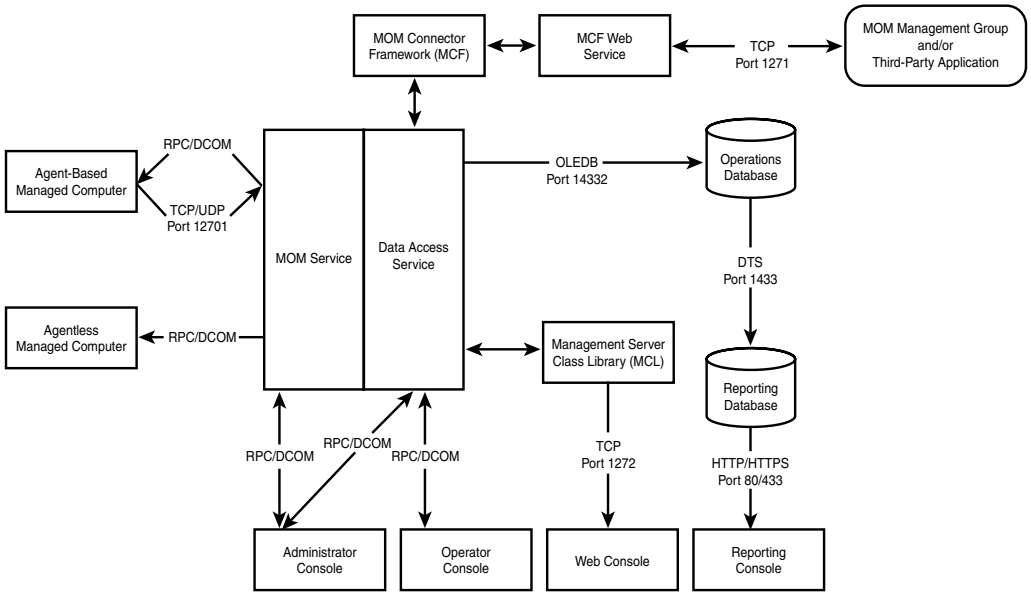

FIGURE 3.4 Component communications protocols and ports.

In keeping with its commitment to the Trustworthy Computing Initiative, Microsoft does not support communications requiring RPC/DCOM across a firewall. Communications are supported which use a standard TCP port that can be secured properly across a firewall, such as the agent-to-management server communications. Table 3.1 lists the various connections, their communications method, and their firewall supportability.

TABLE 3.1 Communications and Firewall Compatibility

From	To	Firewall?	Port, Protocol, or Remark
Agent	Management server	YES	TCP/UDP port 12701
Management server	Agent	NO	RPC (TCP Port 135) and DCOM Ports (TCP/UDP 1024-65535)
Management server	Agentless	NO	RPC (TCP Port 135) and DCOM Ports (TCP/UDP 1024-65535)

TABLE 3.1 Continued

From	To	Firewall?	Port, Protocol, or Remark
Administrator console	Management server	NO	RPC (TCP Port 135) and DCOM Ports (TCP/UDP 1024-65535)
Operator console	Management server	NO	RPC (TCP Port 135) and DCOM Ports (TCP/UDP 1024-65535)
Reporting console	Reporting database	YES	HTTP Port 80 or HTTPS Port 443
Web console	Management server	YES	TCP port 1272
Management server	Operations database	YES	OLEDB Tunneling, port 14332
MOM-to-MOM connector	MOM-to-MOM connector	YES	TCP Port 1271
Connector	Third-party application	YES	TCP Port 1271
Operations database	Reporting database	NO	DTS (TCP Port 1433)

Notice that the agent-to-management server communication method is supported over a firewall, but the management server-to-agent communication method is not. The process of "push" installing agents on managed computers requires RPC and DCOM, whereas the monitoring and rules distribution use a secure TCP port. The downside of this is that if you want to manage an agent on the other side of a firewall, you will have to manually install the agent. Thereafter, the agent will securely initiate the communications. Also, note that managing agentless computers across a firewall is not supported, due to the RPC/DCOM requirements.

The port used by the management server for communicating with agents (12701 by default) is easily configurable on a management server by management server basis. This is also true for the connector port (1271) and the Web console port (1272). You can change the other ports with varying degrees of difficulty.

As Table 3.1 attests, most of the key MOM 2005 communications such as agents and connectors are supported across a firewall, making MOM 2005 a flexible product that can centrally manage your entire enterprise.

How Does MOM Do It?

MOM's internal design and set of components within the management group contains a number of components and complex connections within its architecture. Understanding this architecture can be daunting, so we will approach it by breaking down the layers and looking individually at each component within those layers.

Operations Manager was designed to allow it to deliver all the features in a way that is easy to understand, flexible to a variety of needs, and cost effective. Logically, think of it as being divided into three fundamental layers:

- The data layer
- The business logic layer
- The presentation layer

Organized into logical layers within a management group, MOM provides a high-performance, fault-tolerant, and scalable operations management architecture. These layers, shown in Figure 3.5, each have components working together to deliver the necessary functionality.

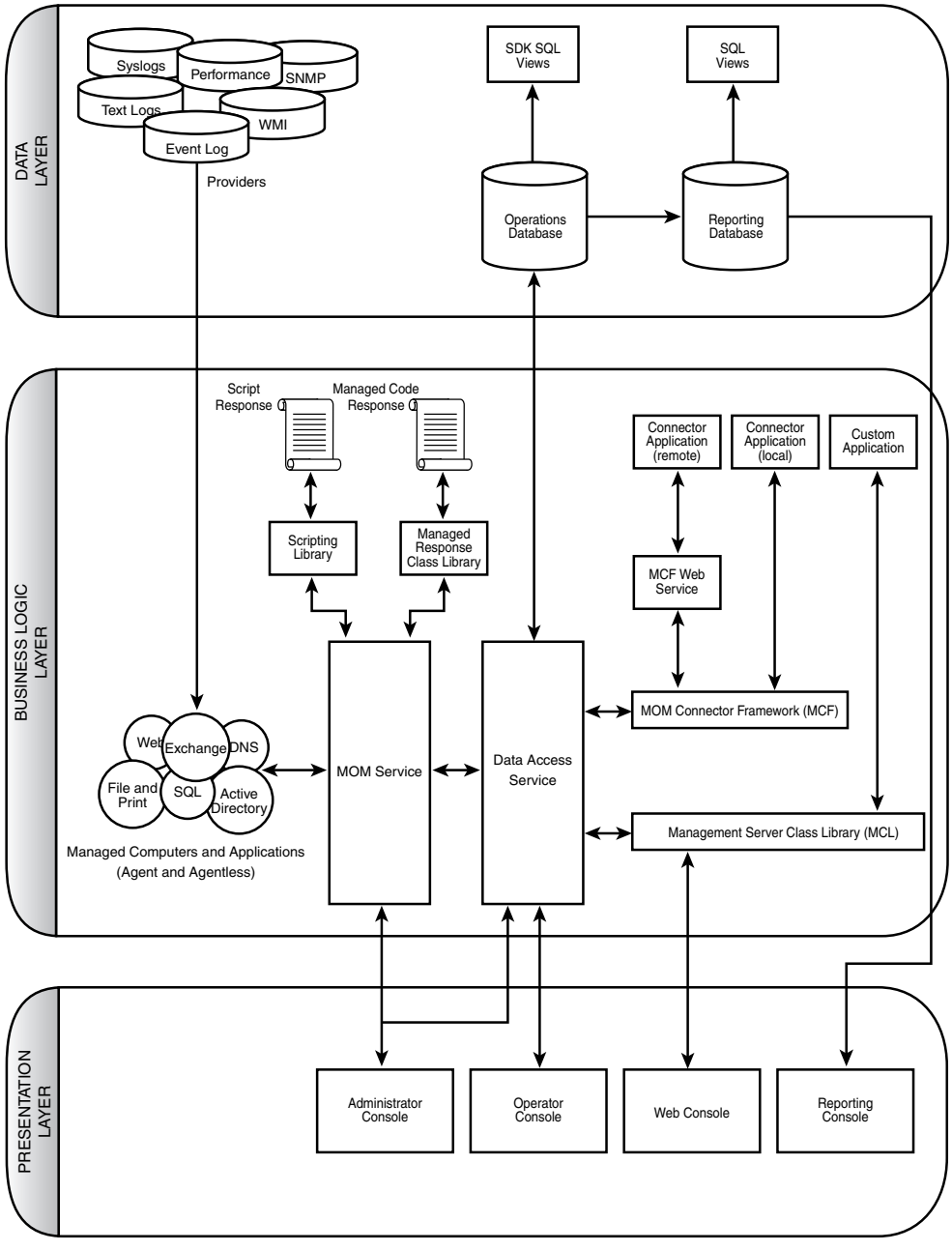

FIGURE 3.5 MOM 2005 architecture and components.

We will look at each of these layers in the following sections of this chapter.

Data Layer

The data layer is the logical layer where the data is stored. This layer is of vital importance to the MOM system, due to the vast quantities of data that need to be received, processed, stored, and acted on. Figure 3.6 shows the data layer and its components.

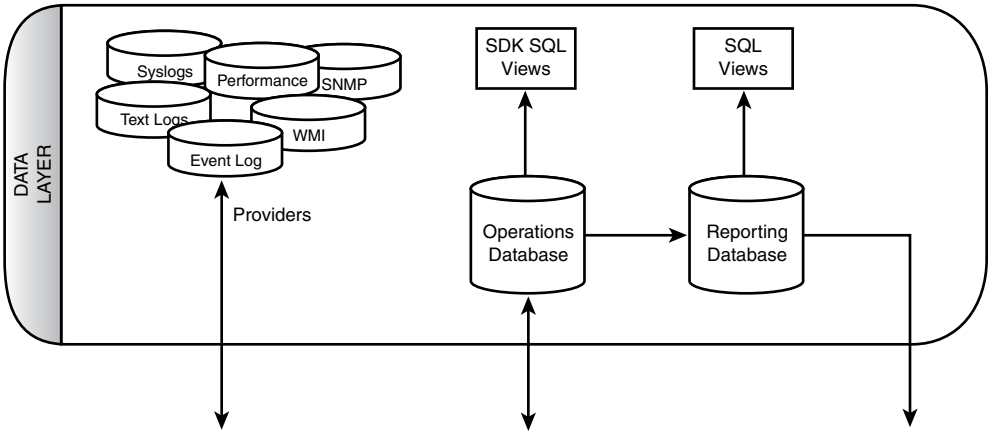

FIGURE 3.6 Data layer and components.

There are five main components within the data layer shown in Figure 3.6:

- Operations database
- Reporting database
- Providers
- SDK SQL views
- SQL views

Without a robust and high-performance data layer, MOM's features would be less effective and could be rendered useless.

Operations Database

The operations database is the centralized repository for MOM's configuration and operational data. This data includes rules, events, performance data, scripts, and the knowledge base. The database engine used is Microsoft SQL 2000 or 2005, either Enterprise or Standard edition. This database is named OnePoint. The name is a holdover from the product's roots before Microsoft acquired MOM and ensures backward compatibility with

MOM 2000. Microsoft plans to change the name of this database in the next version of Operations Manager (see Chapter 23, "Touring Operations Manager 2007," for details).

The OnePoint SQL database is stored in two files by default: the primary database file (EEADATA.MDF) and the transaction log file (EEALOG.LDF), as shown in Figure 3.7. Within the database are more than 350 tables containing the data and configuration settings. Also more than 100 views are defined, providing rapid access to various groupings of the data.

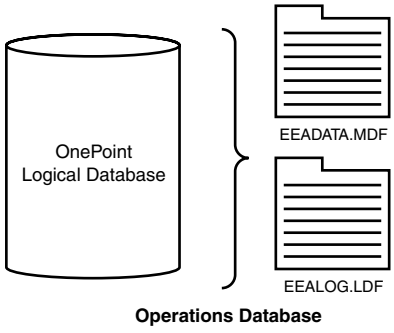

FIGURE 3.7 MOM 2005 database files.

There can be only one operations database in a management group. All collected data, alerts, and configuration data for the management group are stored in the database. This can be a lot of information.

As an example, if the system is collecting data on 100 computers and there are 100 performance counters on each computer measured at 15-minute intervals, there will be close to a million data points collected in a 24-hour period or close to 365 million data points in a year. For 1,000 computers, that would be 3 billion data points per year—and that is just for performance data and does not take into account data from event logs, synthetic transactions, and so on. The sheer quantity of information can be staggering.

The maximum supported size of the MOM 2005 database is 30GB. This limitation is a pragmatic one, in that there is no inherent hard stop limitation built into the MOM code. Exceeding 30GB will not cause the system to halt or fail immediately. The problem is that as the database becomes larger, certain processes such as database grooming take so long to complete that functionality becomes impaired. In the case of database grooming, the database is locked and will not accept additional data while the grooming is taking place. Therefore, management servers collecting data have to buffer the information they have collected and stop accepting new data from agents while the database is locked. Agents then buffer their data while the management server is not accepting data. This can result in delayed alerts because the centralized alerts will not trigger while data is buffered on

the agents. The larger the database, the longer the lockdown window and the more delayed the alerts can get. In a worst-case view, the agent buffers might start overflowing and then information will be lost. Thus, placing an official limit on the database size allows the internal database procedures to complete promptly and the system to function properly.

Real World—Is 30GB Actually a Limitation?

Although 30GB could be considered a limitation, it generally turns out not to be. Our experience with monitoring medium-sized multinational organizations with approximately 250 monitored servers revealed that the operations databases usually held steady within a range of 1.5GB to 2GB. This was using default configuration settings and a standard mix of Microsoft technologies including Windows 2000, Windows Server 2003, Systems Management Server, Exchange 2000 and 2003, and Microsoft SQL Server 2000.

In practice, our conclusion is that the 30GB limit is not going to be a problem for most organizations. Very large organizations monitoring more than 1,000 computers or with heavy monitoring requirements might need to groom more aggressively or increase the number of management groups.

To get around these inherent limitations, MOM includes many features to help maintain the database. Several SQL jobs run automatically to assist in keeping the database trim. The grooming process removes event and performance data that have aged out according to the database grooming setting. Other jobs, created as part of the database installation, perform routine integrity and reindex processes to ensure that the database is healthy and performing well. With the exception of the grooming job, these are standard maintenance jobs that can be performed on any SQL Server database. One job not configured as part of the setup process is the database backup. We discuss procedures for backing up OnePoint and other MOM components in Chapter 12, "Backup and Recovery."

Although the grooming process takes place on a daily basis by default, the actual grooming window is set in the MOM Administrator console under Global Settings. The grooming interval defaults to four days, meaning that events and performance data points older than four days are removed from the operations database when the *MOMx Partitioning and Grooming* job runs at 12:00 a.m. Before the data disappears forever, it is transferred to the reporting database for long-term storage, which we discuss in the next section.

The database is also optimized to allow the grooming to take place quickly, using database partitioning. The database is divided into daily partitions (shown in Figure 3.8). The database is in effect logically broken into daily segments. Grooming and other database-intensive operations can be performed on the logical segments, rather than against the entire database. Most of these operations have specific time constraints, such as grooming data every four days by default or auto-resolving information alerts in four hours. Partitioning allows the database to efficiently retrieve and process the appropriate data.

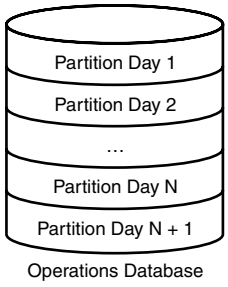

FIGURE 3.8 Operations database partitions.

Reporting Database

The reporting database contains data archived from the operations database. This database is variously referred to as the reporting database, the data warehouse, the archive, and the *System Center Data Warehouse* (SCDW), but the actual SQL Server database name is SystemCenterReporting. The acronym SCDW is frequently used when naming processes within MOM 2005 Reporting, which uses the capacities of the System Center Reporting Server.

As a SQL Server database, the reporting database is stored in two physical files by default: the primary database file (REPDATA.MDF) and the transaction log file (REPLOG.LDF). The database contains more than 100 tables with data and configuration settings. Also more than 300 views are defined, giving the system rapid access to various groupings of the data. These are more views than in the operations database, which makes sense because the reporting database is intended to present information. These views are described in Appendix D, "Database Views."

Data is transferred from the operations database to the reporting database via a Data Transformation Services (DTS) job that runs as a Scheduled Task in the Windows Scheduled Tasks. Similar to the operations database, a job periodically grooms the old data from the reporting database. Both jobs are shown in Table 3.2.

TABLE 3.2 Reporting Jobs

Job Name	Purpose	Default Schedule
SystemCenterDTSPackageTask	Transfers data from the operations database to the reporting database. This job is run as a scheduled task in Windows Scheduled Tasks rather than as an SQL job.	Every day at 1:00 a.m.
SCDWGroomJob	Grooms the SystemCenterReporting database.	Every day at 3:00 a.m.

The grooming interval for the reporting database is one year. The reporting database grooming parameters are hard-coded and buried in a table named WarehouseClassSchema

within the SystemCenterReporting database. The table has a column named WCS_GroomDays that specifies the number of days to groom after, which is 365 for the majority of the data types. The table is keyed on the class ID of the data, so it is not straightforward to modify the information in this table and likely not supported. In the future, Microsoft will provide a user interface method to change these values in a supported manner. Chapter 8, "Post-Installation Tasks," provides information on scheduling the grooming jobs.

Within the one-year grooming window, the reporting database provides a historical view of the operations of your monitored servers. This information is available using reports generated with SQL Server Reporting Services (SSRS).

There is a tentative limit of 200GB for the reporting database, but this is not likely to be the true upper-end boundary. The reporting database growth really only impacts the time needed to generate reports, which does not impact operations functions such as alerting. As the database grows, the database can be separated into different disk subsystems for better performance and even placed on a Storage Area Network (SAN) type technology for performance and growth.

Providers

One of MOM's key advantages is its capability to collect data from a wide variety of sources. This data can be numeric or textual. The information can even reflect missing items that should have occurred within some time frame but did not. This flexibility in the sources of data that MOM can collect and respond to is a key feature in that it allows you to monitor almost anything. For example, many popular brands of Uninterruptible Power Supply (UPS) devices include hardware additions that measure external temperature and humidity. This can be logged to text files or accessed via an Application Program Interface (API). MOM can be configured to read the API or text file, capturing the data and alerting you when the humidity in the server room gets too high or too low.

These sources of data are called *providers*. Provider types include

- **Application logs**—These include the standard event logs, Internet Information Server (IIS) log files, SQL trace log files, ASCII log files, and even UNIX syslog files.
- **Timed events**—These events are generated by MOM and are useful for launching scripts on a regular basis or detecting missing events.
- **Windows Management Instrumentation (WMI) events**—This is a flexible provider, giving MOM access to a wide variety of event-based information through the WMI interface.
- **WMI numeric data**—Similar to the WMI events, this provider gives access to numeric or performance data through the WMI interface.
- **Generic**—This is another class of provider generated by MOM. The Generic provider includes information such as agent heartbeat or events internally generated by scripts.

MOM 2005 includes nearly 700 different predefined providers with its default management packs. You can easily create new providers as needed. Management packs, which are essentially collections of business logic, usually add providers when imported into MOM.

Database Views

MOM includes a number of documented SQL views to help you create custom reports and transfer data from the MOM operations database to other applications and data stores. These views provide read-only access to the MOM database. If you need both read and write access, you can utilize the MOM Windows Management Instrumentation (WMI) classes or the MOM Managed code Application Programming Interface (API), both of which are documented in the MOM Software Development Kit (SDK). The SDK can be accessed at http://go.microsoft.com/fwlink/?LinkId=50272.

The SDK SQL views for the operations database and SQL views to access reporting detail are documented in Appendix D.

Business Logic Layer

The real intelligence of MOM lies in the business logic layer and includes a number of components. Within this layer, rules are set that govern what the business wants to monitor, to be alerted to, to report on, and other myriad details. The business logic layer (shown in Figure 3.9) is where the knowledge of how platforms such as Windows Server 2003 and applications such as Exchange 2003 should be configured and operate are integrated into MOM's framework.

The business logic layer is the most complicated and has the most components of all the layers. The components of this layer are organized into three major groups:

- Core functionality
- Complex responses
- Connecting to external systems

Collecting information from a wide variety of sources is a key characteristic of MOM. This handling of information includes storing it, correlating it to other information, and using it to form alerts and other actions. MOM 2005 collects, handles, analyzes, and responds to operational information using the following components:

- Managed computers and applications
- MOM Service
- Data Access Service (DAS)

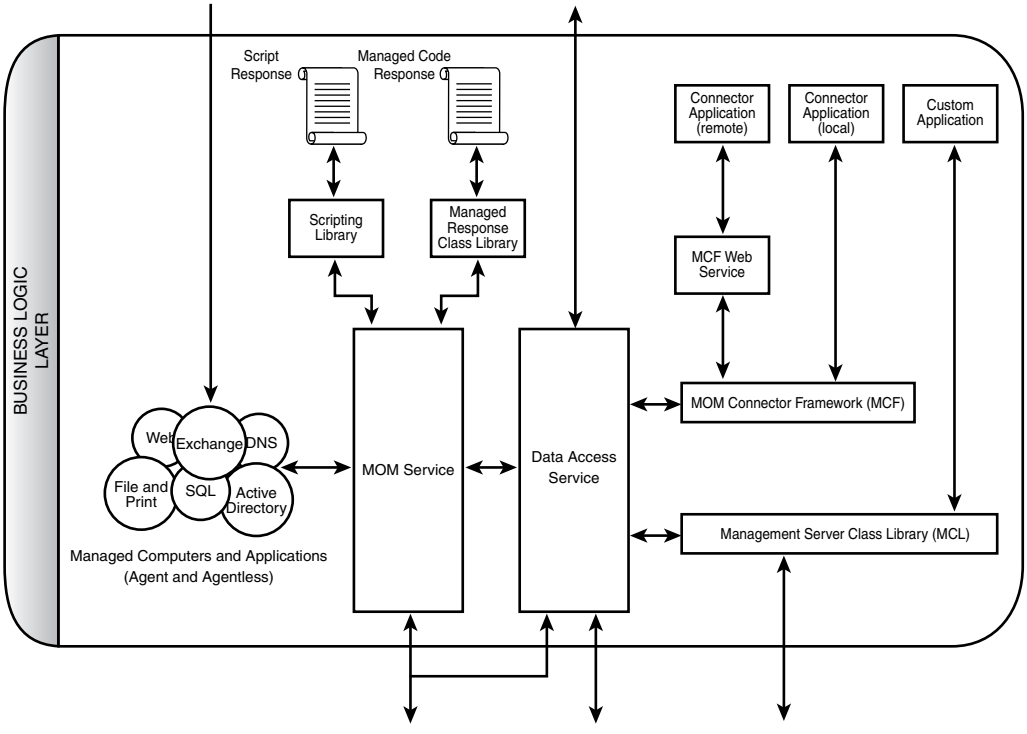

FIGURE 3.9 Business logic layer and components.

The business logic behind a response or task is often more complex or nuanced than even a sophisticated user interface such as MOM's can facilitate. Those cases make it critical that the management application provides a mechanism allowing complex scripts or programs to execute the appropriate business logic. This is accomplished programmatically using the following scripting components:

- Scripting library
- Script responses
- Managed response class library
- Managed code responses

Additionally, the capability to connect to external systems and expose the functionality of MOM 2005 is important to the enterprise scalability and interoperability of MOM. Most large organizations have other management systems or trouble-ticketing systems that are vital to the organizations' operations, and MOM 2005 can interoperate with them. The following components deliver this functionality:

- MOM Connector Framework (MCF)
- Connector applications (local)
- MOM Web Service
- Connector applications (remote)
- Management Service Class Library (MCL)
- Custom applications

We will cover each of the components that make up the functionality of the business logic layer in the next sections. Chapter 19, "Interoperability," discusses the implementation of many of these components.

Managed Computers and Applications—Agent-Based

The method MOM uses for delivering and executing business logic is critical to its success. MOM's agent-based technology allows it to push the work of executing the business logic down to the managed servers. Intelligent local agents are one of the keys to MOM's success. These local agents operate independently, allowing them to respond quickly to changing conditions. The agents are functional even if they cannot contact their management server due to a network outage.

MOM typically manages systems using an installed agent on those systems, although MOM 2005 also includes the capability for agentless monitoring of a small number of systems. The agent runs as a service named the *MOM Service* and collects information as directed by the business logic. The agent is in effect the foot soldier of the MOM system, following the orders dictated by the business logic. The collected information is stored in a buffer locally on the monitored system and then forwarded to a management server. Forwarded information is compressed and encrypted to reduce the footprint on the network and to ensure confidentiality of the management data, allowing MOM to work across slow links and within insecure environments.

Agents can also be configured to look for more than one management server for redundancy and separation of data. Agents can report to a maximum of four management servers, the first of which is assigned automatically by the management service doing the agent install. The other management servers are listed in the agent configuration for automatic fault tolerance. You can see this represented in Figure 3.10, where the agent has a solid line to its primary management server (Monarch) and a dotted line to the other management server (Keystone). The database server is running on a server named Fountain, and the reporting server is Silverthorne.

Normally, the agents are distributed between management servers to provide load balancing as well. This is indicated in Figure 3.11, where Agent 1 reports to Monarch as its primary with Keystone as the backup. However, Agent 2 reports to Keystone as its primary with Monarch as the backup. This balances the management load across the management servers.

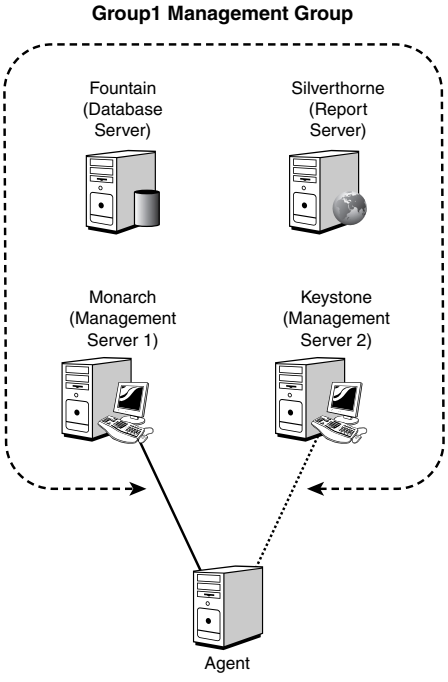

FIGURE 3.10 Agent fault tolerance within a management group.

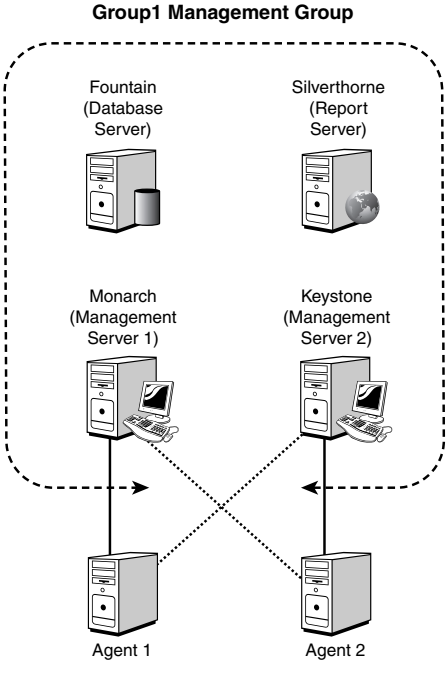

FIGURE 3.11 Agent load balancing within a management group.

Agents can report to multiple management groups as well, which might have different rule sets. This is illustrated in Figure 3.12, where the agent is reporting to both Group1 and Group2. In both management groups, the Monarch management server is primary, and the other management server provides fault tolerance. When reporting to two or more management groups, the agent knows which rules were deployed by which management group and sends the information collected to the appropriate management server. These agents are known as *multihomed*. Chapter 9, "Installing and Configuring Agents," discusses the installation and configuration of agents, including multihomed agents.

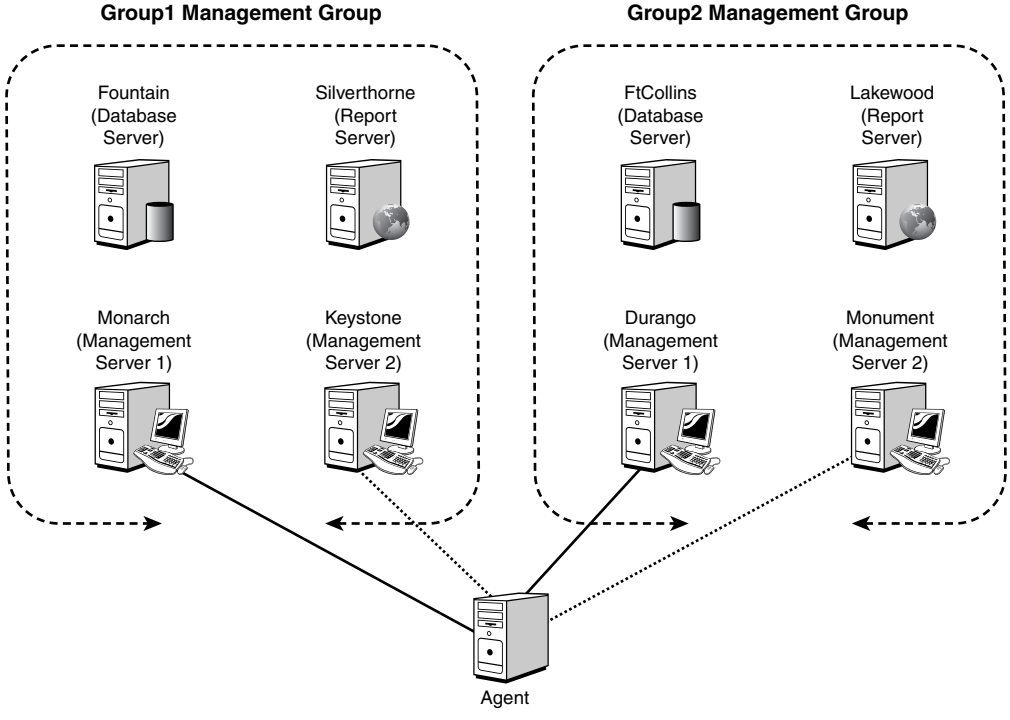

FIGURE 3.12 Agent reporting to multiple management groups.

The agent does not just store and forward the information it gathers to the management groups it reports to. If the business logic dictates, the agent can also evaluate and respond to the information. The responses can be generating an alert, running a script, sending a Simple Network Management Protocol (SNMP) trap, and so on. Having the agents respond locally to events ensures that the system will continue to be monitored and respond to events even when it cannot reach the rest of the MOM infrastructure.

Agents periodically heartbeat to their configured management server. The heartbeat information is used to ascertain whether the agent is still alive and save the management server from having to poll for the status of all its agents. The heartbeat information also contains the latest agent configuration version information. Right after a heartbeat, the

agent also uploads any queued data to the management server. The management server then uses the configuration version information to send the latest configuration information to the agent, ensuring that the agent has the latest business logic applied to it. More information on agent heartbeats is available in Appendix A.

Agent Processes—MOM Service and MOM Host

Although we have been referring to *the agent*, things are actually a bit more complicated than that. The MOM agent actually uses two processes to achieve its objectives: the MOM Service and MOM Host processes, shown in Table 3.3. The MOM Service process handles the internal workings of the agent and communications with the management servers. The MOM Host process handles the information gathering and responses that the business logic dictates. There may be multiple instances of the MOM Host process on any given managed computer. The agent runs multiple MOM Host processes to ensure that the response and providers are isolated. If the process locks running a script or retrieving data from a provider, it will not affect the function of the overall agent or the function of any other MOM Host process.

TABLE 3.3 MOM Agent Processes and Tasks

Processes	Executable	Tasks
MOM Service	MOMService.exe	Communication with management server(s)
		Applications event log—Read/Write
		Security event log—Read/Write
		WMI event provider—Read
		File transfer—Send/Receive
MOM Host	MOMHost.exe	Monitors and collects Windows event log data
		Monitors and collects Windows performance counter data
		Monitors and collects WMI data
		Monitors and collects Application log data
		Runs script and batch responses
		Runs managed code responses

Even with gathering all this information and taking the appropriate actions, the footprint on the monitored system is light. On a typical managed system, agent activities consume less than 1% of processor time. The memory requirements will vary according to number of business logic rules applied to the system, but a rough estimate is between 20 and 60 megabytes. Even the agent on the management servers, where the agent service is handling events from a number of systems, uses less than 1% processor time as an average. Depending on the number of systems managed by a particular management server, the processing time scales with the number of events forwarded and can grow to 5% to 10% of processor time.

MOM can actually tell us what the overhead is. In Figure 3.13, you can see the results for a typical MOM installation monitoring approximately 20 systems. For clarity and simplicity, we selected only three systems for graphing. The graph measures the percentage processor time over the last seven days for the MOM Service process on three systems, including the management server (Monarch). What might not jump out at you is that the scale of the graph is 0% to 1% (the axis values scale automatically based on the values generated). You can see that for most servers the time is less than one-tenth of a percent (.1%). It is interesting to note that in the graph you can see that each of the managed computers has a consistent load, which is proportional to what is being monitored on the server. The exception is the management server itself, which is hovering around 0.25% with spikes of up to about 0.75%. The spike represents the extra work MOM does every day at 2:00 AM to look for new computers. There are also several spikes on 7/12/2006 at about 11 AM and 8 PM, which could bear some investigation. However, even during the spikes the management server load is less than 1% of average processor utilization.

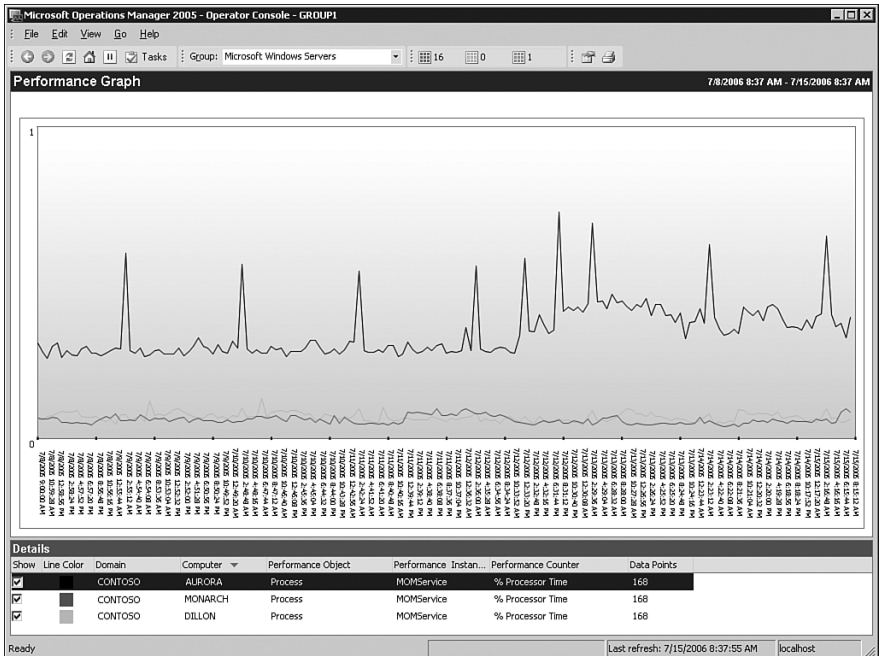

FIGURE 3.13 Typical MOM Service processor utilization.

MOM Service

The memory requirements of the MOM Service process are much more variable, being proportional to what is monitored or the number of rules deployed to the managed computer. Figure 3.14 shows the results of the agent memory utilization over a seven-day period. The roles of the servers are definitely relevant here. The Monarch computer is the MOM 2005 management server, Dillon is a computer running mainly IIS and

antivirus/antispam software, and the Aurora server is running Exchange 2003. The agent on Dillon monitors relatively basic and static functions, so the memory utilization holds steady over the entire seven days at a bit above 6.6 million bytes or about 7MB. The agent on Aurora shows even less variability in the memory utilization but uses more memory and hovers just at under 20MB. This is an Exchange 2003 server and is one of the more managed roles within MOM 2005, hence the higher memory requirements. The agent on Monarch does a variety of other tasks that we will cover in the next section ("MOM Host"), so the memory utilization is much higher and grows from 59MB to just under 66MB.

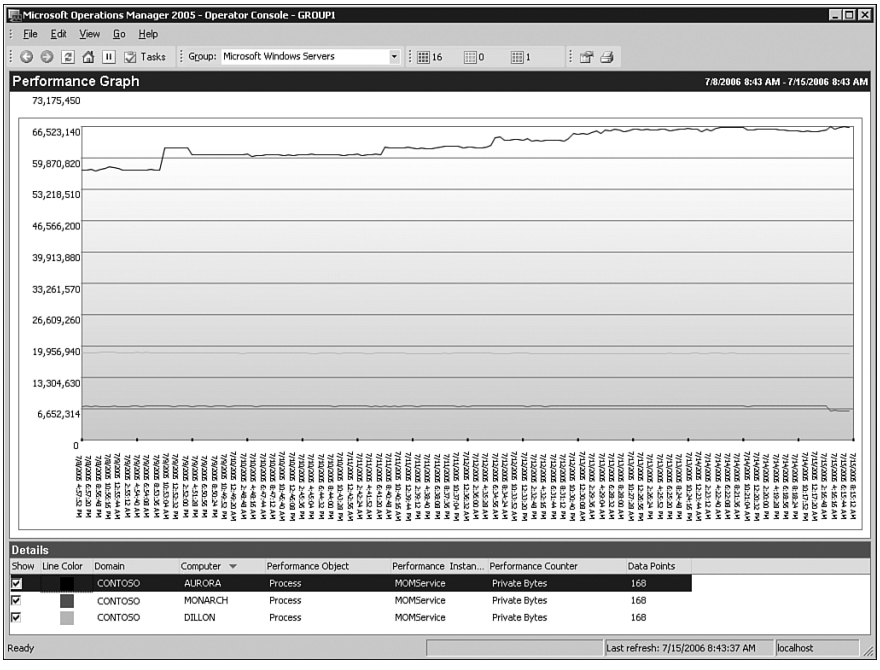

FIGURE 3.14 Typical MOM Service memory utilization.

MOM Host

The other process, the MOM Host, has its own processor and memory utilization patterns. Its processor and memory requirements are separate and in addition to those of the MOM Service process. In Figure 3.15 you can see the processor utilization for the MOM Host processes on the same three computers. The first thing to notice is that there are two instances of the process (MOMHost, MOMHost#1) for each managed computer and in fact there could be more in some cases depending on the different responses and providers that the agent is monitoring. However, you can see that over the course of the seven days of monitoring the requirements for the servers are quite light. For all the servers, the MOMHost instance is close to zero processor utilization, and the MOMHost#1 instance is less than 0.25%. This holds true even for the management server itself.

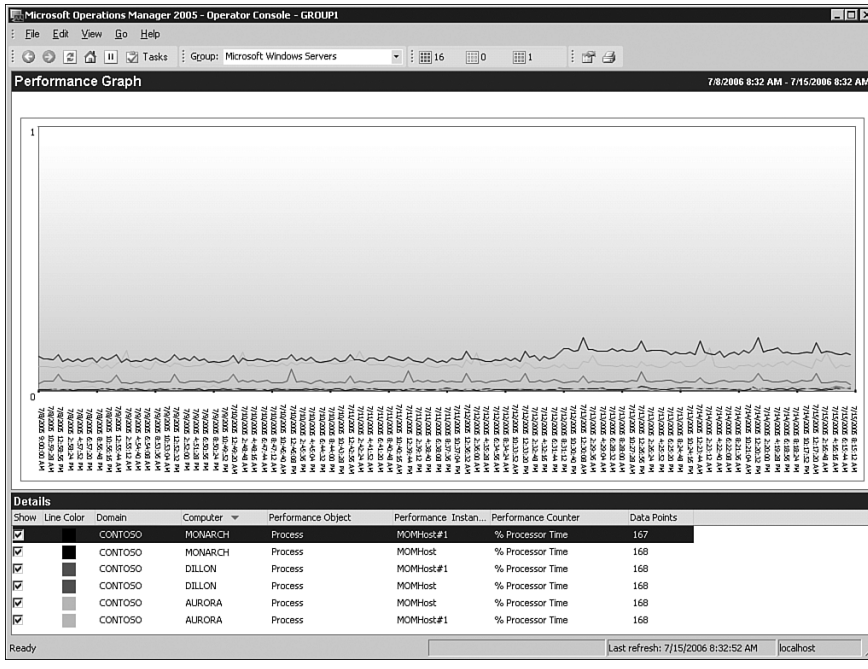

FIGURE 3.15 Typical MOM Host processor utilization.

MOM Host memory utilization tends to be more activity driven than the processor utilization and so exhibits a bit more variability, at least for the second instance of the process (MOMHost#1). The memory requirements again vary by the class of the server and thus by the monitoring requirements. In the case of MOM Host, this is more directly proportional to the level of monitoring. As expected by this, you can see in Figure 3.16 that for the Aurora Exchange server the first instance, the MOMHost process holds steady at just under 10.6MB (or about 11MB), and that the second instance (MOMHost#1) is at just under 21MB at the end of the monitoring period. For the Dillon managed computer, the MOMHost instance is steady at about 7MB, and the MOMHost#1 instance varies between 4MB and 5MB. Finally, the Monarch management server MOMHost instance is steady with a jump midway in the monitoring period at about 8MB, and the second instance, MOMHost#1, hovers right at about 11MB (though there is a brief drop in the middle of the monitoring period). That jump in MOMHost reflects the addition of an agentless managed computer to the management server, which we will discuss in the next section of this chapter.

With all that detailed information, a good question to ask is how much processor and memory resources will be utilized by the agent process overall? As was shown in the previous discussion, it varies by the level of monitoring and the class of server. Our sample computers fall into the categories of a lightly monitored web server, a heavily monitored Exchange server, and a management server. Tables 3.4 and 3.5 show the total processor and memory utilization for the various agent processes in this typical environment.

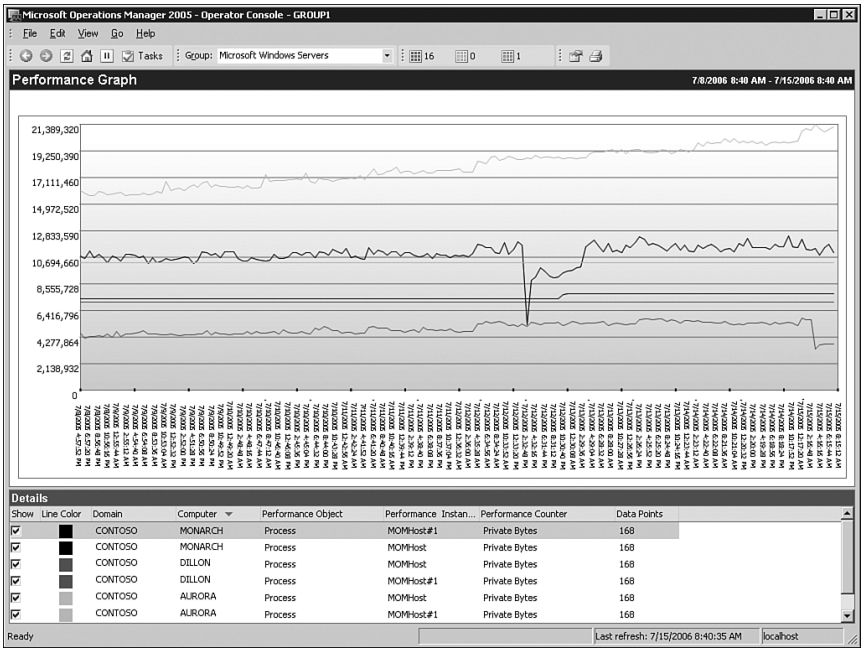

FIGURE 3.16 Typical MOM Host memory utilization.

TABLE 3.4 Typical Agent Memory Utilization

Class	Computer	MOMService	MOMHost	MOMHost#1	Total Memory
Web Server	DILLON	7MB	7MB	5MB	19MB
Exchange Server	AURORA	20MB	11MB	21MB	52MB
Management Server	MONARCH	66MB	8MB	11MB	85MB

TABLE 3.5 Typical Agent Processor Utilization

Class	Computer	MOMService	MOMHost	MOMHost#1	Total Processor
Web Server	DILLON	0.10%	0%	0.25%	0.35%
Exchange Server	AURORA	0.10%	0%	0.25%	0.35%
Management Server	MONARCH	0.75%	0%	0.25%	1.00%

These numbers are not significant, given current memory standards for servers. Even for a web server with only 256MB of RAM, the total agent memory utilization of about 19MB is about 7% of total memory. For a typical Exchange server with 1GB of RAM, the 52MB memory utilization is about 5% of total memory. Even for a typical management server with 1GB of RAM, the 85MB memory utilization is less than 8% of total memory. Far more is used by the SQL Server service! The processor numbers speak for themselves. Table 3.6 summarizes the typical resource utilization on managed computers and management servers.

TABLE 3.6 Typical Utilization Summary

Class	Memory Profile	Processor Utilization	Memory Utilization
Web Server	256MB	0.35%	7%
Exchange Server	1GB	0.35%	5%
Management Server	1GB	1.00%	8%

Managed Computers and Applications—Agentless

In addition to the primary method of managing computers using agent-based technology, MOM 2005 is capable of managing computers without an agent. In this mode, the management server agent takes on the role of the agent for the agentless managed computer. The information is gathered remotely using RPC and DCOM.

The information gathered is equivalent to that for agent-based managed computers. However, agentless monitoring has the following limitations:

- **Scalability**—The work of monitoring agentless managed computers is done by their management server and more specifically by the agent on the management server. This load is significant and can severely impact the performance of the management server, so the agentless mode is limited to only 10 agentless managed computers per management server and a total of 60 agentless managed computers per management group. This is a major limitation to the scalability of the agentless mode of operation.
- **Tasks**—With no local agent on the managed computer, tasks cannot be executed locally on the managed computer. Tasks that run on the management server can be executed against the agentless managed computer, such as the Ping task.
- **Event log descriptions**—Event descriptions are not gathered as part of the agentless monitoring, so the event log descriptions are not available unless the management server has the same event log messages .DLL file. An awkward workaround is to install the same software on the management server as is on the agentless managed computer.
- **Firewall support**—Given that the management server uses RPC and DCOM to monitor the agentless managed computer, agentless managed computers are not supported across a firewall. Opening up RPC and DCOM across a firewall cannot really be done securely, given the nature of the protocols.
- **Management pack limitations**—Management packs presuppose agent-based managed computers and may not fully operate on agentless managed computers. Most of the monitoring functions will work without any problems, but many of the more sophisticated responses will not function correctly.

Given these limitations, the agentless monitoring features are not suitable for many tasks, and this is definitely not the method to manage the majority of computers. However, in some cases such as those where there is concern about the installation of software on an

application server, this will be a solution allowing a reasonable level of monitoring with a limited level of impact to the managed computer. It helps capture those one-off computers that normally resist management.

Agentless Monitoring for Windows NT4 Systems

One use for agentless monitoring is to monitor Windows NT4 systems, as the MOM 2005 agent is not supported on Windows NT4.

MOM Service Component

The MOM Service component (also known as the MOM Server component) is different from the MOM management server. The MOM Service is a component of a management server that runs as a set of services and handles key functionality, which we discuss in this section. The MOM management server refers to a role within the MOM 2005 infrastructure where there are a collection of components on a given computer, including the MOM Service, the DAS, and so on.

The MOM Service performs the following functions for the management server:

- Manages agent installation
- Manages agent configuration
- Monitors managed computer availability
- Consolidates data
- Monitors server-side responses
- Self monitoring
- Monitors agentless managed computers

Using computer discovery rules, the MOM Service component scans the directory for computers that match the computer discovery rules. After discovering computers, the MOM Service component can initiate an agent install and update for the soon-to-be-managed computers. This installation can be configured to take place automatically, or it can require administrative authorization before proceeding. The default is for the system to require approval for agent installations and to wait for 48 hours before removing agents from a system that falls out of the managed computer rules.

In addition, the managed computer attributes are scanned to discover what applications are installed on a managed computer and what roles they play. The attribute scan process is done via a task, which is executed by the agent rather than remotely by the management server. However, the MOM Service component initiates that task and receives the results back from the agents.

Attribute information is used to assign the computer to computer groups within MOM. The MOM Service component is also responsible for scanning computer group

memberships. Membership in computer groups is based on attributes the agent discovers, as well as explicit assignments made at the console. Additional information on computer attributes and group membership formulas is available in Chapter 13, "Administering Management Packs."

Automatic rule deployment and view selection take place after the computer is placed in the appropriate computer groups. Based on the computer group membership, the MOM Service component delivers the appropriate rules to the agents on the managed computers. In this regard, it is the sergeant of a MOM system, passing on the orders for the agents to follow. It is also natural for the business logic and thus the rules that represent that logic to change. These rule updates are automatically distributed to the appropriate managed computers by the MOM Service component.

The agents periodically check in, or heartbeat, to their management server. The MOM Service component receives that heartbeat and also detects when a managed computer has not generated a heartbeat. It is during this heartbeat process that agents check to see whether there are new rules or rule updates that they need to receive from the MOM Service component.

The MOM Service component receives operational data from the agents and passes it on to the DAS component, in effect operating as a proxy between managed computers and DAS. The MOM Service component not only proxies the operational data but also processes the operational data and executes responses indicated by the business logic.

The MOM Service component also performs the agent functions for the management server that it runs on, so you will not find a separate service for the agent on a management server. Finally, the MOM Service component acts as the agent for agentless managed computers. This includes polling of agentless managed computers, collecting the operational data remotely, and running responses (where possible).

Processes Used by the MOM Service Component

Similar to the agent on a managed computer, the MOM Service component uses two processes to achieve its objectives: the MOM Service process and the MOM Host process. The nomenclature is somewhat confusing because there is a *MOM Service component* and a *MOM Service process*. The MOM Service component is composed of two processes, one of which is the MOM Service process. See Figure 3.17 for a graphical view of this.

The tasks, shown in Table 3.7, differ somewhat from agent tasks. The MOM Service process handles the internal workings of the local agent, communications with the agents, and passing the collected information to the DAS component. It also processes the rule updates sent to the agents. The MOM Host processes handle the information gathering and the responses that the business logic dictates for the local computer and for the agentless managed computers. MOM Host also handles the agent installs and uninstalls and updates the configuration settings on the agents.

TABLE 3.7 MOM Service Component Processes and Tasks on the Management Server

Processes	Executable	Tasks
MOM Service	MOMService.exe	Communicates with agents (receiving data and updating rules)
		Relay agent data and rules to and from the DAS component
		Applications event log—Read/Write
		Security event log—Read/Write
		WMI event provider—Read
		File transfer—Send/Receive
MOM Host	MOMHost.exe	Installs and uninstalls agents on managed computers
		Updates agent configuration
		Monitors and collects Windows event log data (local and agentless)
		Monitors and collects Windows performance counter data (local and agentless)
		Monitors and collects WMI data (local and agentless)
		Monitors and collects application log data (local and agentless)
		Runs script and batch responses (server-side, local, and agentless)
		Runs managed code responses (server-side, local, and agentless)

The MOM Service process handles the bulk of the communications with the agent-based managed computers, initiated by the managed computer agent. In contrast, the MOM Host process manages the majority of the communications with the agentless managed computers, and the communications are fundamentally initiated by the management server. This is shown in Figure 3.17.

Data Access Service Component

The Data Access Service (DAS), also known as the Data Access Server, handles both data insertions and data requests to the MOM database. It handles all insertions to the operations database. It handles most of the requests for data as well. The most important exception to this is the reporting subsystem, which bypasses the DAS and uses DTS to transfer data from the operations database to the reporting database. However, the DTS process only copies the data and does not remove data from the operations database.

The DAS is a server-based Component Object Model Plus (COM+) application hosted by the DLLHOST process. The DAS exposes a set of DCOM objects and communicates that control access to the MOM database. The DCOM interfaces are associated with COM+ roles and provide authentication and authorization of the identities that access the interfaces.

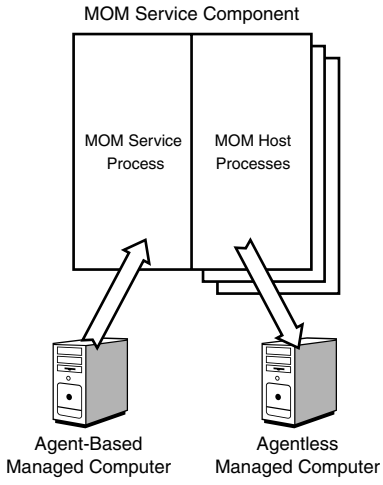

FIGURE 3.17 MOM Service component, processes, and managed computers.

Rather than being installed as a service, the DAS is installed as components in Component Services for improved performance by using object pooling. More than 100 different components are within the DAS. The DAS provides common centralized database access logic, centralized query logic, shared cache, and pooled connections to the operations database server. This translates to improved performance for the database server in reduced connections and duplicate requests, and also for the other components in reduced latency when retrieving cached information. Due to the centralized access and query logic, there is also less likelihood of data being entered incorrectly by wayward components. All components see the same consistent view of the data and operate under a consistent security model.

Some critical services the DAS provides are maintaining data consistency and logging. Whenever a change is made, such as the updating of an alert, the DAS records the change and the credentials of the user making the change, storing it in the operations database as well. This is important for auditing and security.

Programmatic Response Components

The scripting capabilities of MOM 2005 allow for customized monitoring and responses to events, alerts, and performance data.

There are two major types of programmatic responses in MOM 2005:

- **Script responses**—These allow you to extend the capabilities of the basic rules. Scripting responses are flexible and easy to use. MOM supports its own scripting interface with VBScript or JScript, or you can use custom scripting languages such as PerlScript. The scripts are stored within the operations database and are visible and editable from the MOM Administration console.

Another advantage of script responses is that the code and any updates delivered to the managed computer by the management server and agent use the rule delivery process. Scripts can be executed on either the agent or the management server, as appropriate.

- **Managed code responses**—This response type can call a method within a .NET Framework assembly. These assemblies can be developed in any .NET Framework-compliant language such as Visual C# .NET, Visual Basic .NET, Visual C++ .NET, Visual J# .NET, and so on.

  The assemblies are not delivered to the managed computer by the MOM infrastructure, so they require manual distribution and updating.

  A big advantage of the managed code response type is that it can call practically any .NET Framework assembly. These calls can be made either on the managed computer or the management server.

The programmatic responses are supported by two libraries:

- **Scripting library**—The COM-based scripting class library contains various runtime scripting objects, such as the Alert object, Event object, PerfData object, and ScriptContext object. These objects allow response scripts to interact with alerts, events, and performance data. See Chapter 22, "Using and Developing Scripts," for more information.
- **Managed response class library**—The managed code response class library is a .NET Framework class library and is equivalent to the COM-based scripting objects in the scripting library.

  The Microsoft.EnterpriseManagement.Mom.Runtime namespace contains classes and other types for creating MOM-managed code responses. The items in this namespace are defined in the MOM.Context (MOM.Context.dll) assembly.

Table 3.8 compares the capabilities of the two programmatic response types. Overall, script responses are generally easier to use and are better integrated into the MOM 2005 infrastructure. Managed code responses require more effort to create and deploy but perform faster and have a wider array of application interoperability.

TABLE 3.8 Comparison of Programmatic Responses

Feature	Script Responses	Managed Code Responses
Programmatic access to the response context.	Yes	Yes
Capability to create a new alert.	Yes	Yes
Capability to create a new state monitoring alert.	Yes	Yes
Capability to create a new MOM event.	Yes	No

TABLE 3.8 Continued

Feature	Script Responses	Managed Code Responses
Capability to create a new MOM performance data item.	Yes	No
Capability to create and submit computer discovery data.	Yes	No
Supported programming languages.	All COM-compatible scripting languages such as VBScript and JScript, including third-party extensions such as PerlScript	All .NET Framework languages, including third-party extensions
The programming language of the response must be explicitly specified.	Yes	No
Capability to store response as native code.	No	Yes
Stored in MOM database.	Yes	No
Capability to directly invoke an application component from a MOM rule.	No	Yes
Distribution mechanism.	Management Packs	Manual
Deployment mechanism.	MOM agents	Manual
Update mechanism.	MOM agent updates	Manual
Source code can be viewed in the MOM User Interface (UI).	Yes	No
Source code can be edited in the MOM UI.	Yes	No
Capability to call COM components.	Yes (Limited)	Yes
Capability to call .NET assemblies.	Yes (Limited)	Yes

We will look at script responses in more detail in Chapter 22.

Connecting to Other Management Platforms

MOM 2005 is not alone in the enterprise. In a typical enterprise Information Technology (IT) ecosystem, there might be several other management applications ranging from trouble ticket systems to management frameworks. MOM 2005 is designed to integrate with those systems. The integration is fundamentally at the alert level, which supports the following functionality:

- Sending new MOM alerts to external applications
- Sending MOM alert updates to external applications
- Receiving alert updates from external applications to the MOM system
- Receiving new alerts from external applications to the MOM system

In other words, the alert flow is bidirectional. MOM alerts can be forwarded to the other management application and kept in sync on both platforms as the alert changes. Alerts generated in the external management application can be inserted into the MOM database and also kept in sync. For example, if an alert is forwarded to a trouble ticket application, resolving the alert in either the MOM console or the trouble ticket console results in the alert being resolved in both consoles.

In fact, this is the method that MOM 2005 uses to communicate between management groups in a complex MOM hierarchy.

The components that provide the functionality listed previously are

- MOM Connector Framework (MCF)
- MOM Web Service
- Connector applications (local)
- Connector applications (remote)

The MOM Connector Framework is a managed .NET class library that provides an infrastructure for developing connector applications. The MCF manages the communication of alerts and alert updates between MOM 2005 and the connector application. The MCF provides business logic to support the development of custom connectors between MOM and other management applications. The MCF is accessible as both a standalone class library and a web service. Connector applications running locally on the management server can access the class library, and connector applications running remotely need to use the web service. The MCF provides support for connector applications running on non-Windows platforms such as UNIX through the MCF Web Service.

Although similar functionality can be achieved through developing custom applications using the Management Service Class Library (MCL) discussed in the next section of this chapter, there are several advantages to developing connector applications using the MCF. These advantages include the following:

- **Tracking alerts**—The MCF handles the details of tracking which alerts have been forwarded and which ones require updating. This tracking means that the connector application does not need to include the code and logic for those functions, which simplifies the development effort.
- **Crossing firewalls**—The MCF Web service uses port 80 and can easily cross firewalls if needed. The SSL protocol can be used to increase security, in which case port 443 is used, which also crosses firewalls.
- **Alert knowledge**—The MCF also provides easy access to the alert's product knowledge content, if that needs to be forwarded with the alert.
- **Bidirectional logic**—The MCF has built-in logic to handle bidirectional synchronization. The MCF makes it easy to develop two-way connectors, which keep alerts synchronized.

The MCF also allows alert suppression and other logic to be handled using the standard MOM rules. The connector application will not need to perform these tasks, providing better integration into the MOM 2005 infrastructure with less development effort. The rules are stored in management packs and are easily configured by MOM administrators with no need for development skills or controls.

Two namespaces are defined within the MCF. The Microsoft.EnterpriseManagement.Mom.Connector is included for backward compatibility with MOM 2000 SP1. The Microsoft.EnterpriseManagement.Mom.Connector.V2 is the more feature-rich version for MOM 2005 and can also be exposed via a web service on the management server. See the MOM 2005 Software Development Kit (SDK) at http://go.microsoft.com/fwlink/?LinkId=50272 for more information.

Management Service Class Library (MCL) and Custom Applications

The MOM Management Server Class Library (MCL) is used to develop custom applications that access the MOM 2005 operations database programmatically using Visual Studio .NET. The MCL is a .NET Framework class library that exposes MOM operations data, configuration information, and information about the rule hierarchy. This class library is only available on management servers, meaning that the custom applications that use the class library must be developed, tested, and run on the management servers.

The Microsoft.EnterpriseManagement.Mom namespace defines the general-purpose classes and types for accessing MOM operations data, rules, and computers. Items in this namespace are defined in two separate assemblies:

- The Microsoft.Mom.SDK Assembly (in Microsoft.Mom.Sdk.dll)
- The MOM.Context Assembly (in MOM.Context.dll)

Custom applications that use classes in this namespace should reference both assemblies in the Visual Studio .NET project. These assemblies can be found in the SDK Bin folder of the MOM program files folder and in the management server's Global Assembly Cache.

For more information on the Management Server Class Library and its usage, see the MOM 2005 Software Development Kit (SDK), available at the link referenced previously.

Presentation Layer

The last layer of the Microsoft Operations Manager 2005 system is the presentation layer (shown in Figure 3.18), which allows the information gathered by MOM to be viewed and the system to be controlled.

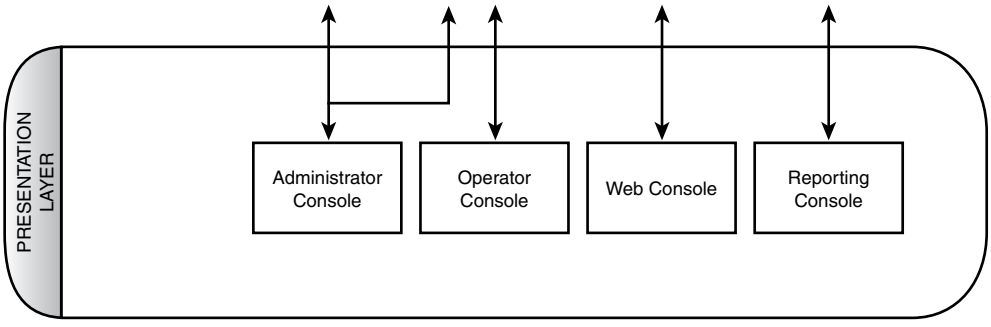

FIGURE 3.18 Presentation layer and components.

Four console components and a reporting subsystem component present the data and provide administrative and operational control of the MOM 2005 system. The consoles are as follows:

- Operator console
- Web console
- Administrator console
- Reporting console

The first two consoles (Operator and Web consoles) support the operations functions for MOM users. These consoles target the operators or IT administrators who actually manage systems and applications. The Administrator console contains the tools used to configure MOM 2005 and is targeted at the MOM administrator. The Reporting console provides access to operational information, both at a detailed operational level and at the broader managerial level. This console is targeted at a range of users from operators to managers and even to executives.

The presentation layer is arguably the most important layer because it is the layer that users of the system interface with. Without a solid presentation layer, the system cannot be used to its maximum potential.

Operator Console

The Operator console presents the operational view of the infrastructure and applications. This is the console in which the most time will be spent by the operators (that is, most users) of MOM 2005, hence the name of the console. This console is roughly modeled on the same lines as the Microsoft Outlook user interface, utilizing the research that Microsoft has invested in developing user interfaces to allow users to view information and complete tasks. It is designed to allow an operator to quickly and successfully handle operational events by doing the following:

- **Identifying**—Knowing that an event has occurred is half the battle in IT. The Operator console rapidly identifies and presents the relevant events and information. It also prioritizes them automatically.
- **Understanding**—Just knowing that an event has occurred is not enough for the operator to be successful. Understanding what an event means, both in the context of other events and with detailed knowledge, is the critical next step in the process. The Operator console delivers in-depth understanding to the operator of the context around the event, detailed knowledge about the event itself, and the event and computer history.
- **Resolving**—Finally, the problem that the event represents needs to be resolved. The Operator console puts both the detailed knowledge of possible solutions and the tasks to execute those solutions at the fingertips of the operator. It can even launch those solutions automatically, while informing the operator of actions taken.

The Operator console is organized to support role-based operators, such as SQL Server administrators, Exchange administrators, or enterprise administrators. The roles can be defined by technology, location, or any other logical grouping. The Operator console allows administrators to monitor and troubleshoot the servers and applications under their responsibility.

The MOM 2005 agents gather a wide variety of information about the managed computers, ranging from configuration to events to performance data. The Operator console displays that variety of information through a number of view types, presented in the following list and discussed more fully in Chapter 8:

- Alerts
- State
- Events
- Performance
- Computers and groups
- Diagram
- Views (Public Views and My Views)

These views can be manipulated in a multitude of ways within the Operator console. The console allows the information to easily perform the following functions:

- **Navigation**—Moving through the various views and the individual items such as alerts and events is a key function of the Operator console. The console is organized to allow operators to easily navigate between views and levels of detail, as well as be able to move back in the same way a web browser would.

- **Scoping**—A big problem with any console is managing the large number of computers, events, and alerts. The Operator console simplifies this by allowing you to set the scope to any computer group, which then narrows the scope of everything that is displayed. For example, if an Exchange administrator selects the Microsoft Exchange Installed Computers group in the console, all the administrator will see as he navigates is the information specific to that group. The groups can be defined geographically or functionally, so a group can be created to follow your IT functional groups and thus narrow the scope as well.

  Even though the scope is adjusted, all the information for all the managed computers is still available in the console should the administrator need to broaden the scope. On the flip side, users can be restricted to their functional scope using Console Scope so that they can see only the information relevant to the group of computers they are responsible for.

- **Drill down**—The Operator console provides summarized views of information but allows you to drill down into the details and back out again. For example, you can click on a state icon in the State view to see the state component view, double-click on the state component to drill into the alerts related to that component, and then drill into the events that generated the alert. You can easily jump all the way back out or go back one level by clicking the Back button.

  Given all the information, complex interrelations, and level of detail inherent in the operational data, the drill-down Operator console makes it easy to navigate through the sea of information.

- **Execute tasks**—The Operator console allows you to easily execute context-sensitive tasks no matter where you are in the console. You can launch a task by selecting a node in a Diagram view, an alert in the Alert view, or a computer in the Computer view. The tasks will launch based on the identity of the computer represented in the information. The tasks are self-policing, meaning that an Exchange task will not allow you to execute it against a non-Exchange computer. This helps prevent unexpected consequences.

The Operator console is an application built on the .NET Framework and is not an MMC console, a departure from the normal Microsoft standard. This is in great part due to the complexity of the UI and the need to present the information in a flexible and fast manner.

The console is organized into four panes composed of the Results Pane and three additional panes, which are the Navigation Pane, the Details Pane, and the Task Pane. The panes are shown in Figure 3.19 and discussed in the following list:

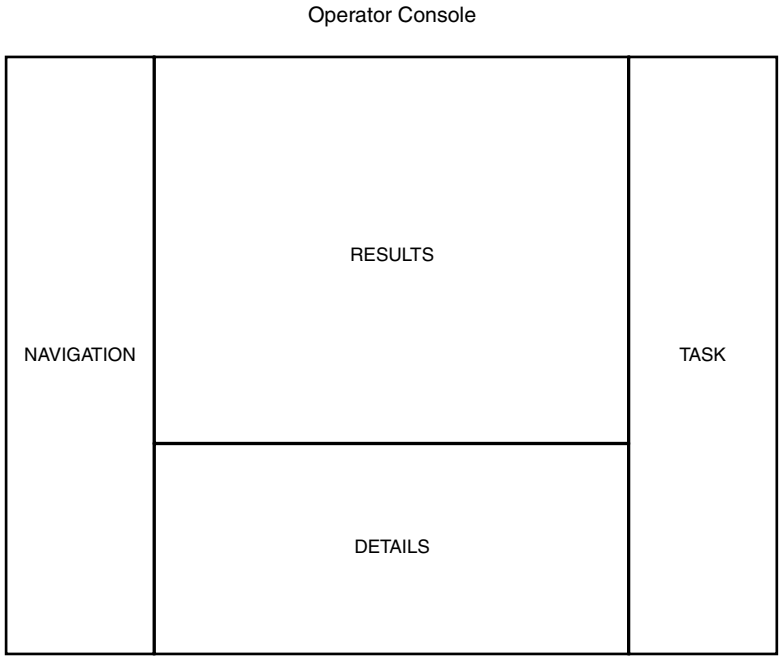

FIGURE 3.19 Operator console organization.

- **Results Pane**—The Results Pane displays the results of a view selection (such as alert or diagram), filtered by the computer group selected. This pane displays the lists of alerts, events, computers, diagrams, and so on—that is, the information that the view is presenting.
- **Navigation Pane**—The Navigation Pane shows a console tree for the currently selected view and the view selection buttons in the bottom of the pane. When a view button is clicked, the console tree adjusts to the appropriate tree, and the active Results Pane changes to reflect the view.
- **Details Pane**—The Details Pane shows the details for the item selected in the Results Pane. There is no Details Pane shown for the Diagram view. The Details Pane is organized by tabs to provide easy access to a wealth of information. For example, the Alert view Details Pane has tabs for Properties, Custom Properties, Events, Product Knowledge, Company Knowledge, and History.
- **Task Pane**—The Task Pane shows a tree view of the tasks and task folders. These tasks are not filtered by context, though they are grayed out if they do not apply to the selected computer.

You can choose not to display the Navigation, Details, and Task panes simply by selecting View within the Operator console and then deselecting the appropriate panes. You can also display up to three different results panes at a time in the Operator console, each

with a different view selected. This is useful if you need to be viewing the state, alerts, and events all at the same time. It is particularly useful if you have a large screen with high resolution!

Web Console

The Web console is a scaled down console that is accessible via HTTP and provides stripped-down versions of the Alerts, Computers, and Events views of the Operator console. It gives computer operators and application owners an easy-to-use and simplified console from which they can check the operational status of their assigned computers or applications. The Web console provides a basic MOM console to a larger group of administrators that manage a narrow set of servers or applications, requiring less training and without installation on their desktops.

As shown in Figure 3.20, you can view alerts, events, and computers, and even change the resolution state of the alert from within the Web console. The console provides full access to any knowledge associated with an alert, which supports the full range of identifying, understanding, and resolving alerts.

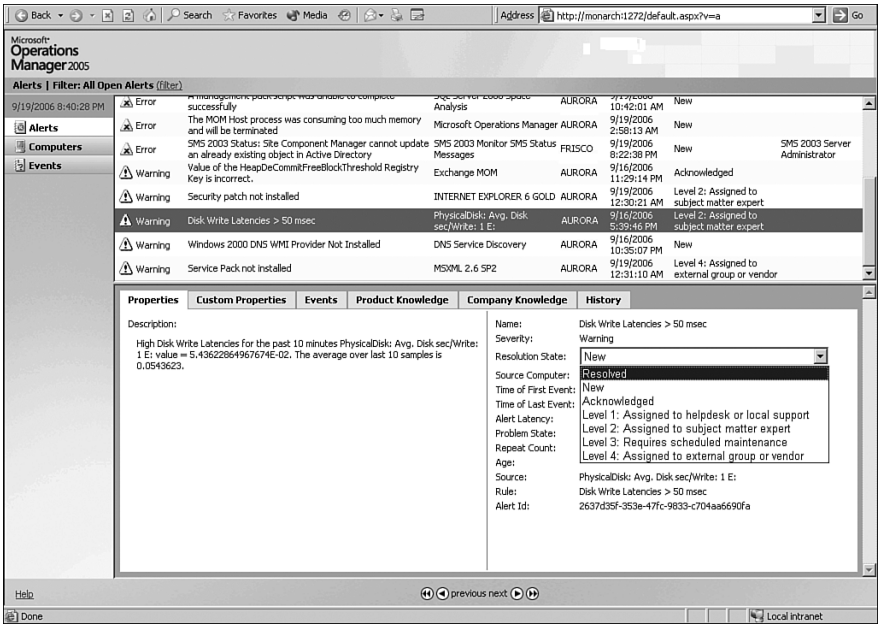

FIGURE 3.20 Web console.

When email notifications are triggered by alerts, the link in the email is a link to the Web console. The default address of the console is http://*<management server name>:1272*. Both the address and the port can be changed within the Administrator console. The Web console can also be launched from within the Administrator console. The default port for the HTTP access is 1272. The port can be changed, and SSL can be used for security.

The Web console is a good example of a custom application that leverages the Management Server Class Libraries (MCL).

Administrator Console

The Administrator console is used to configure and administer MOM 2005 itself. This is the console where the MOM administrators will spend most of their time. In contrast with the Operator console and in keeping with Microsoft standards, the Administrator console is an MMC snap-in.

The console has two panes—the Navigation Pane and the Detail Pane. The Navigation Pane presents a folder type view of the available options. The Detail Pane presents a variety of different types of information and options depending on the particular section being viewed.

The Administrative console is organized into four major segments:

- **Information Center**—The information center provides links to jump to information about MOM 2005, including the MOM 2005 website, downloads, documentation, technical support, licensing, and security. These links allow you to quickly get to the most recent literature and management packs.
- **Operations**—The operations segment presents links to launch the other consoles, including the Operator console, the Reporting console, and the Web console.
- **Management Packs**—This is the segment of the Administrator console where the business logic is maintained. The computer groups, rule groups (or management packs), tasks, notification and operators, scripts, computer attributes, and providers are all accessible from this segment.

  When you click on the Management Packs node in the Navigation Pane, you will see a summary of the business logic in the Detail Pane, as well as links to the other nodes. The business logic summary gives you a count of the rule groups, management pack rules, custom rules, computer groups, and number of scripts.
- **Administration**—The administration segment of the console is the area from which agents are deployed, the mode for managed computers is set, console administration is defined, global settings such as database grooming are defined, and MOM connects to other systems. The sections of the administration segment are Computers, Console Scopes, Global Settings, and Product Connectors.

  When you click on the Administration node in the Navigation Pane, the Detail Pane displays a summary of the MOM 2005 management group architecture and managed computers, as well as links to the other nodes. The management group summary includes the number of management servers, agent-managed computers, agentless managed computers, unmanaged computers, cluster computers, and total number of computers.

What you see in the Navigation Pane of the Administrator console, and the actions that you can take, are determined by your security access. The Administrator console will only display the nodes for the segments in which you have rights. If you have not been granted access to the business logic, the management pack node will not be displayed. If you have not been granted administrative access to the management group, the administration node will not display. MOM security is discussed further in Chapter 11, "Securing MOM."

Reporting Console

The Reporting console is not really a MOM 2005-specific console but rather is the Microsoft SQL Server Reporting Services console. SQL Server Reporting Services is a central and feature-rich solution that enables creating, managing, and delivering both paper-oriented reports and interactive web-based reports from almost any data source, including the SQL Server database. SSRS combines the data management capabilities of SQL Server and Microsoft Windows Server with Microsoft Office components to deliver useful reports.

SSRS supports the full reporting life cycle, including

- **Report authoring**—Report developers can create reports to be published to the Report Server using Microsoft or third-party design tools that use Report Definition Language (RDL), an XML-based industry standard used to define reports.
- **Report management**—Report definitions, folders, and resources are published and managed as a web service. Managed reports can be executed either on demand or on a specified schedule and are cached for consistency and performance.
- **Report delivery**—SSRS supports both on-demand (pull) and event-based (push) delivery of reports. Users can view reports in a web-based format or in email.

The Reporting console allows you to view published MOM 2005 reports, manage security for access to the reports, and manage subscriptions to the reports. Reporting and particularly report creation is a complex topic and will be addressed in detail in Chapter 21, "Using and Developing Reports."

Summary

This chapter introduced the MOM 2005 architecture and data flow. We also discussed components that will be referenced throughout this book. The material in this chapter should help in planning your installation and deployment of MOM. The next chapter discusses the process of planning your MOM deployment.