

CHAPTER 12

Backup and Recovery

All production systems should have established backup and recovery procedures in place, and an Operations Manager (OpsMgr) infrastructure is no exception. Out-of-the-box, OpsMgr 2007 does not include a backup process. If one of the databases becomes damaged through corruption or a hardware failure and you are without its backup, you will have to reinstall that component and re-create the database. If there is damage to the Root Management Server (RMS) or Operational database, you will have to reinstall the entire management group. This creates all kinds of headaches.

Re-creating a database without the ability to restore what was previously there means that you lose all the information in the database. In the case of the Operational database, you lose all customization and operational data collected in the database. If you have installed the Reporting Component and you lose the Data Warehouse and ReportServer databases, you lose the reporting data you have accumulated in the data warehouse, plus OpsMgr report definitions and report customizations, which are stored in the ReportServer database. If you install Audit Control Services (ACS) and lose the ACS database, you have lost your security logs and audit information.

IN THIS CHAPTER

- ▶ Roles of Key OpsMgr Files and Databases
- ▶ Establishing a Backup Schedule
- ▶ Database Grooming
- ▶ Backing Up and Restoring the SQL Server Databases
- ▶ Backing Up the RMS Encryption Keys
- ▶ Recovering from a RMS Loss
- ▶ Backing Up Management Packs
- ▶ Backing Up Reports
- ▶ Backing Up SQL Reporting Services Encryption Keys
- ▶ Disaster Recovery Planning

There are also critical files that you need to secure through backup. As an example, the RMS includes encryption keys. If the RMS is damaged and the encryption keys cannot be recovered, you will have to build a new management group and RMS. (Microsoft alleviates this situation somewhat with Service Pack (SP) 1, see the “Recovering from a RMS Loss” section later in this chapter.) The Reporting Server Component also has encryption keys. These types of potential data loss make it critical to create a backup and recovery plan for your OpsMgr 2007 implementation.

This chapter discusses backup and recovery strategies for Operations Manager. It also looks at a methodology for handling large report databases and requirements for disaster recovery planning.

Roles of Key OpsMgr Files and Databases

Backing up appropriate files and databases in a timely manner facilitates minimal data loss if there is a catastrophic failure in your OpsMgr infrastructure. An Operations Manager installation includes system databases, user databases, and significant files that you will want to protect from data loss.

SQL Server System and User Databases

Microsoft SQL Server *system* databases include databases established during the database engine install. These databases are integral to the functionality of the database engine, and include the master, msdb, model, and tempdb databases. Other databases, created for application-specific purposes, are *user* databases.

Operations Manager-specific user databases include the Operational database, Data Warehouse database, and ACS database. Installing the SQL Server 2005 Reporting Component (required for the data warehouse) creates two additional databases: the ReportServer and ReportServer tempdb databases.

Note that the Operations Manager 2007 setup process allows you to specify database names for the three databases it creates. This chapter will refer to the default names.

You should include the following items in your backup strategy. This includes various system and user files and databases:

- ▶ **The Operational database (named *OperationsManager* by default)**—This is Operation Manager’s database installed for each management group, and is the most important database to back up. If you lose this database due to a hardware failure or corruption and do not have a database backup, you will have to reinstall the RMS and re-create the database, losing all rule customizations, discovery rules, and operational data collected. This database is shared among management servers within a management group, and must be backed up for every OpsMgr management group.

- ▶ **The Data Warehouse database (*OperationsManagerDW* by default)**—This database stores aggregated data used for reporting, which is used by SQL Reporting Services (SRS) for trend analysis and performance tracking. Based on the amount of data you are collecting and the degree of aggregation, this database may be large and thus require special handling. If you have not installed OpsMgr Reporting, your management group does not include the *OperationsManagerDW*, *ReportServer*, or *ReportServerTempDB* databases.
- ▶ **The SQL Reporting Services *ReportServer* database**—This database is used by the SQL Reporting Services Component. It stores the report definitions used for OpsMgr Reporting and is updated when new reports are defined or definitions of existing reports are changed.
- ▶ **The *ReportServerTempDB* database**—The only reason to back up *ReportServerTempDB* is to avoid having to re-create it if there is a hardware failure. If there is a hardware failure, you do not need to recover the data in the database, but you will need the table structure. If you lose *ReportServerTempDB*, the only way to get it back is by re-creating the SQL Reporting Services *ReportServer* database.
- ▶ **The ACS database (named *OperationsManagerAC* by default)**—This database is associated with the Audit Collector service, which runs on the ACS collector. The database uses an agent to track cleared Security Event logs, and adds a new table daily for each day's security events. If you have multiple collectors, each uses its own ACS database.

ACS typically uses its own instance of SQL Reporting Services and the SQL Reporting Services database, in which case you will also need to accommodate these items in your backup strategy. Chapter 15, "Monitoring Audit Collection Services," includes a full discussion of ACS.

- ▶ **The Master database**—This is a system database, recording all information used by a SQL Server instance—including database file locations, configuration settings, and security and login account information. This database should be backed up whenever there is a change to your SQL Server configuration. If you installed the Operations, Data Warehouse, Reporting, or Audit database Components on separate database servers or instances, each will have a Master database that should be backed up. This is also true for a separate database server or instance using SRS.
- ▶ **The Msdb database**—The Msdb database is also a SQL Server system database, containing scheduled tasks information for jobs, including regularly scheduled database backups. If you have installed the Operations, Data Warehouse, Audit database, or SRS Components on separate servers, each server will have a Msdb database that should be backed up.

- ▶ **Management packs and reports**—Management packs contain rules and information pertaining to how Operations Manager monitors applications, services, and devices. The management packs are stored in the Operational database, which you should back up as part of your standard procedure. We recommend separate backups of non-sealed/customized management packs because this provides the granularity to import them directly into Operations Manager if necessary and to save a self-contained copy of any rule customizations. Instances of importing management packs could include rolling back changes to an unsealed management pack or moving a customized management pack from a development to production environment.

Report templates are stored in the ReportServer database. As with management packs, we recommend separate backups of any reports you have created or customized.

- ▶ **IIS metabase**—Both the Web Console Server and SRS components use Internet Information Services (IIS). Most IIS settings are saved in its metabase, although several settings are in the Registry. If you are running IIS 6.0 with Windows Server 2003, the IIS metabase is automatically backed up each time the in-memory database is written to disk. The backups are saved to `%SystemRoot%\System32\inetsrv\History`.

To create your own metabase backups, see <http://support.microsoft.com/kb/32477> for IIS 6.0 or <http://support.microsoft.com/kb/300672> for Windows 2000 / IIS 5.0. The IIS 5.0 metabase backups, which must be performed manually, are stored at `%SystemRoot%\system32\inetsrv\MetaBack`. The IIS backup files can be saved for disaster recovery using a physical disk backup.

- ▶ **Custom files**—Custom files include encryption key files for the RMS and Reporting Server components. Customizations to console views are saved in the local user profile on the computer running the console. Those personalizations could be backed up with physical disk backup or a SystemState copy of the local operating system.

Establishing a Backup Schedule

In addition to identifying required files for backup, you should also establish a regular backup schedule. Tables 12.1 and 12.2 give suggested time frames for backing up significant databases and files used by Operations Manager 2007.

Establishing a daily backup schedule for those files that change regularly helps ensure any data loss affects less than 24 hours worth of data. It also makes it possible to meet your Service Level Agreements (SLAs) if you have backups of the information necessary to restore any OpsMgr components!

TABLE 12.1 OpsMgr Databases with Recommended Backup Schedule

Database	Name	Type of Data	Recommended Backup Schedule
Operations database	OperationsManager (default)	This database contains the majority of the OpsMgr configurations, settings, and the current operations data. The loss of this database would require completely reconfiguring Operations Manager and result in the loss of all operational data.	Daily
Data Warehouse database	OperationsManagerDW (default)	This database holds all the data used for reporting and can be large. The loss of this database would mean the loss of all historical operations and reporting data.	Daily or Weekly
SQL Reporting database	ReportServer	This database holds all the report definitions, as well as cached report information and snapshots. The loss of this database would mean having to reimport reports and re-create subscriptions. Minimal impact. If you have installed ACS, it will have its own reporting subsystem and instance of the ReportServer database.	Monthly
ACS database	OperationsManagerAC (default)	This database tracks Security Event logs being cleared by an agent. A new table is created daily for that day's events.	Daily
Master database	master	This database is a SQL system database and records the system information for SQL Server 2005. Back up the Master database for every SQL Server instance in your OpsMgr environment.	Daily or when changes

TABLE 12.1 Continued

Database	Name	Type of Data	Recommended Backup Schedule
Msdb data-base	msdb	This database is a SQL system database and holds information on all jobs scheduled through SQL Server. It can be found on every SQL Server instance in your OpsMgr environment. Although OpsMgr 2007 does not use this database as it schedules its maintenance internally, if you create any jobs yourself within SQL Server (backups or database maintenance, for example), you should back up the Msdb database to retain that information.	Monthly or as needed

TABLE 12.2 Significant Files with Recommended Backup Schedule

File	Type of Data	Recommended Backup Schedule
Management packs and reports (.mp and .xml files)	Source files for management packs and reports. Enable more granular restoration than entire Operations database; also used for moving management packs and reports from one management group to another.	After changes to management packs or reports
Custom files	Encryption key files, the IIS metabase, and so on.	As needed

Database Grooming

Part of maintaining the integrity of your database environment is managing data retention for your Operational and Data Warehouse databases. Data retention also affects the size of the database and the amount of data to back up, which affects your backup requirements and scheduling.

Grooming the Operational Database

The OpsMgr2007 Operations console includes the ability to modify data retention settings for the Operations database under Administration -> Settings -> General -> Database Grooming. The default setting for each of the data types is to remove or groom the data after seven days (see Figure 12.1). After the data is groomed, it is not recoverable unless it was previously backed up.

Within the Operational database, the `p_partitioningandgrooming` stored procedure runs automatically at midnight to perform the grooming. To run grooming manually, execute this procedure, which calls a series of other stored procedures that use your database grooming settings to perform the grooming.

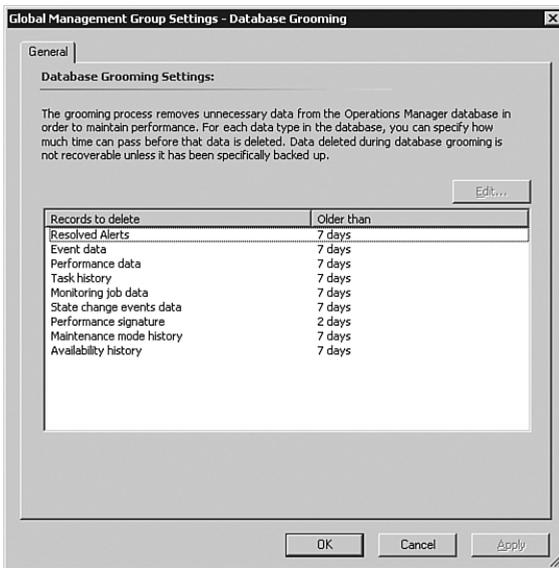


FIGURE 12.1 Operations Manager database grooming settings.

Grooming the Data Warehouse Database

The Operations console does not have a graphical interface to modify data retention settings for the data warehouse. You can groom the data warehouse settings by modifying columns in certain tables inside the `OperationsManagerDW` database. Data is groomed out at different intervals depending on the degree of aggregation. Data is stored by type, and the ranges for data retention vary from 10 days to 400 days by default, depending on the type of data.

Grooming Settings

Microsoft stores grooming-related settings in two areas in the MaintenanceSetting table in the Data Warehouse database:

- ▶ **Instance space**—Discovered objects with their properties and relationships.
- ▶ **Config space**—This is space that contains information about your management packs, rules they contain, overrides you have created, and so on.

Columns of interest and their default values are shown in Tables 12.3 and 12.4.

Using the default settings for the config space, a sealed management pack is removed 400 days after it was uninstalled from all management groups that are members of the data warehouse! This is also true for non-sealed management packs, but OpsMgr retains up to three old versions of a non-sealed management pack as well.

TABLE 12.3 MaintenanceSetting Table Instance Space Settings

Column	Value
LastInstanceGroomingDateTime	The last time grooming operations were performed
InstanceGroomingFrequencyMinutes	Frequency of the grooming process start in minutes (default: 480)
InstanceMaxAgeDays	Maximum age (since the day the instance was deleted) for the instance space objects (default: 400)
InstanceMaxRowsToGroom	Maximum number of objects to delete in one run (default: 5000)

TABLE 12.4 MaintenanceSetting Table Config Space Settings

Column	Value
LastConfigGroomingDateTime	The last time grooming operations were performed
ConfigGroomingFrequencyMinutes	Frequency of the grooming process start in minutes (default: 60)
ManagementPackMaxAgeDays	Maximum age for the management pack (since the day MP was uninstalled) (default: 400)
NonSealedManagementPackMaxVersion Count	Maximum number of non-sealed MP versions to preserve (independent of age) (default: 3)

Data Retention

Settings controlling data retention are located in the StandardDatasetAggregation table. You can view the grooming settings by running the following SQL query:

```
SELECT AggregationIntervalDurationMinutes, BuildAggregationStoredProcedureName,
GroomStoredProcedureName, MaxDataAgeDays, GroomingIntervalMinutes, MaxRowsToGroom
FROM StandardDatasetAggregation
```

Table 12.5 displays the default settings returned by the SQL query.

The following applies to the results shown in Table 12.5:

- ▶ The first column is the interval in minutes that data is aggregated. NULL is raw data, 60 is hourly, and 1440 is daily.
- ▶ MaxDataAgeDays is the maximum number of days data is retained. Depending on the type of data and its degree of aggregation, defaults can range from 10 to 400 days.
- ▶ GroomingInterval Minutes is the grooming process frequency. Performance, Alert, Event, and AEM data is groomed every 240 minutes (4 hours); State data is groomed every hour.

TABLE 12.5 Data Returned from StandardDatasetAggregation Table

AggregationInterval DurationMinutes	BuildAggregation StoredProcedureName	GroomStored ProcedureName	MaxData AgeDays	Grooming IntervalMinutes	MaxRows ToGroom
NULL	NULL	EventGroom	100	240	100000
NULL	NULL	AlertGroom	400	240	50000
NULL	NULL	StateGroom	180	60	50000
60	StateAggregate	StateGroom	400	60	50000
1440	StateAggregate	StateGroom	400	60	50000
NULL	AemAggregate	AemGroom	30	240	100000
1440	AemAggregate	AemGroom	400	240	100000
NULL	PerformanceAggregate	Performance Groom	10	240	100000
60	PerformanceAggregate	Performance Groom	400	240	100000
1440	PerformanceAggregate	Performance Groom	400	240	100000

To make sense of the grooming settings in this table, look at non-aggregated Event data, which is the first row of information in Table 12.5. We know that this pertains to Event information because of the referenced procedure name EventGroom (GroomStoredProcedureName). The information returned from the query tells us that Event data is not aggregated (AggregationIntervalDurationMinutes=NULL) and is saved for 100 days (MaxDataAgeDays). The EventGroom stored procedure grooms data (GroomStoredProcedureName), and runs every 240 minutes/4 hours (GroomingIntervalMinutes). Each time the stored procedure runs, it will groom a maximum of 100,000 rows.

You can use the following SQL code to change the grooming frequency for each type of data:

```
USE OperationsManagerDW
UPDATE StandardDatasetAggregation
SET MaxDataAgeDays = <number of days to retain data>
WHERE GroomStoredProcedureName = '<procedure name>' AND
AggregationIntervalDurationMinutes = '<aggregation interval duration>'
Go
```

Datasets

The data itself is retained by data type. Each data type is stored in a separate structure, called a *dataset*. Examples of these datasets include a performance dataset for performance data, a state dataset to monitor state transitions, an event dataset for events, and so on.

Management packs may also introduce new datasets. All datasets in existence—known today—are referred to by Microsoft as *standard datasets*. Microsoft maintains a set of tables for standard datasets that hold a description of the dataset including its data retention policies. A non-standard dataset does not have to follow the same rules; data retention settings for non-standard datasets are dataset specific.

Data retention for the standard dataset is set at the *aggregation* level, meaning that performance raw data (the samples themselves) is stored a certain number of days. The number of days may differ from the number of days the daily aggregates are stored for performance counters. These settings are stored in the StandardDatasetAggregation table, shown in Table 12.5 in the previous section.

The primary key for the StandardDatasetAggregation table is composite and consists of the database ID (from the Dataset table) and the Aggregation TypeID (from the AggregationType table). Default values will vary by dataset/aggregation type. The aggregation types, defined in the AggregationType table (which consists of the AggregationTypeID, AggregationTypeDefaultName, and AggregationTypeGuid columns), are as follows:

- ▶ **0**—Raw data
- ▶ **10**—Subhourly aggregation
- ▶ **20**—Hourly aggregations
- ▶ **30**—Daily aggregations

For performance reasons, data is not always groomed row-by-row. If the data inflow is high (typically the case for medium and large organizations for performance and event data), the Data Warehouse database uses additional tables to store data. This makes the grooming process (database row deletes) more efficient, as an entire table can be deleted rather than individual rows.

As an example, ten million performance samples are stored in the first instance of a table. After ten million records, OpsMgr creates a new table that holds the additional data and calculates a minimum and maximum date for the data in the first table. This information is stored separately in the StandardDatasetTableMap table. Grooming looks at this table to determine what data exists in each table and grooms accordingly. For OpsMgr 2007 SP 1, Microsoft has announced a ResKit tool named DWDATARP that allows you to view and set the data retention policies for all configured datasets.

The logic used by the grooming process is as follows:

- ▶ For a certain dataset/aggregation type combination, check to see if there is only one table in the data warehouse.
- ▶ If there is just one table, delete records row-by-row using the DELETE TOP SQL statement and MaxRowsToGroom parameter from the StandardDatasetAggregation Table.
- ▶ If there is more than one table, find the table with the oldest Maximum Date for data in it. If this date is older than the retention period, drop the entire table; otherwise, do not delete any rows.

The implication of following this process is that the data warehouse may not always be “current” on grooming. When the data in a table spans a month, some records are kept one cycle or month longer than necessary. However, the performance gains of dropping an entire table versus performing individual row deletes in SQL Server is enormous, so storing the data a little longer seems a reasonable tradeoff. Because report selection criteria includes a time period, any additional data is not visible to the user.

NOTE

How Is Grooming Actually Performed?

There are separate stored procedures to groom different types of data such as Performance, State, Alerts, Events, AEM data, and so on. The GroomStoredProcedure-Name column in Table 12.5 specifies the grooming procedures used for the data warehouse.

You can use the `standarddatasetgroom` stored procedure in the data warehouse database to trigger grooming to happen manually. The procedure uses a parameter, `datasetid`. This value, listed in the dataset table, represents the type of data to be acted on. Steve Rachui documents this at <http://blogs.msdn.com/steverac/archive/2007/12/13/scom-2007-operational-and-datawarehouse-grooming.aspx>. OpsMgr will call the `standarddatasetmaintenance` stored procedure to execute `standarddatasetgroom`.

As OpsMgr aggregates most of the data in the Data Warehouse database, its growth on a day-to-day basis is less than the Operational database. However, since the retention period is longer, it will grow to be considerably larger.

Data Warehouse Backup Considerations

Because the Data Warehouse database has longer data retention periods, it can grow to be quite large, although it is initially smaller than the Operational database. Large databases can present potential backup issues. A terabyte database, as an example, can take a long time to back up and restore.

One approach is to create archived or segmented versions of the Data Warehouse database, separating it by different months, quarters, or years, depending on its size and your reporting requirements. This portioning gives you granularity in backups—once a database is archived, it does not have to be backed up on a regular schedule. It also makes potential restore operations quicker.

A sophisticated backup schedule that accommodates archive databases would back up the current data warehouse (OperationsManagerDW) but retain online copies of archived versions. As you backed up each archived database when it was current, you would simply maintain those tapes in long-term storage as long as required for reporting purposes. Segmenting the reporting information allows you to reduce the volume of data backed up on a daily basis while maintaining long-term availability of the data using archived databases and long-term tape storage.

There are (of course!) several caveats to this:

- ▶ Adjusting the grooming settings (discussed in the “Grooming the Data Warehouse Database” section of this chapter) to groom only at the end at your designated archival period.
- ▶ Administrative overhead in managing the backup process at the end of each retention period. The end-of-period backup process adds complexity, which we illustrate in Figure 12.3.
- ▶ Tailoring your reports to run against archived data as necessary.

For purposes of illustration, consider a company monitoring 2,000 servers. For simplicity's sake, let's assume the company does not have third-party software with a SQL backup agent and uses Microsoft SQL Server's backup capability for its database backups. The company needs access to a year's worth of data. We can use the formula discussed in Chapter 4, “Planning Your Operations Manager Deployment,” to determine the amount of space required for the data warehouse. Plugging these numbers into the formula

$$(3 \text{ MB/day} \times \text{Number of Agents} \times \text{Retention Days}) + 570 \text{ MB} = \text{Data Warehouse size}$$

gives a Data Warehouse database of nearly 2.2 terabytes (TB), which is too large to easily back up directly to tape. In addition, backup to a disk file requires equivalent storage on disk for the backup file, for a total of over 4TB. This is also too much storage for practical operations.

However, data for a single quarter will be just over 500 gigabytes (GB) or one-half terabyte. This amount is within the capability of the tape backup system. The company decides to break up the data warehouse into quarterly archives and accordingly sets the data warehouse grooming to groom data after each quarter (120 days). This configuration has been running for more than a year, so they have a steady state condition.

Figure 12.2 illustrates the backup process. You can see that the current data warehouse is available (OperationsManagerDW), as well as the four previous quarters of archived data (4Q2007, 3Q2007, 2Q2007, and 1Q2007). The process consists of two steps:

- ▶ Perform an online backup of the data warehouse to a disk file.
- ▶ Back up the backup disk file to tape. In the event of a disaster, the tape backup can be easily restored.

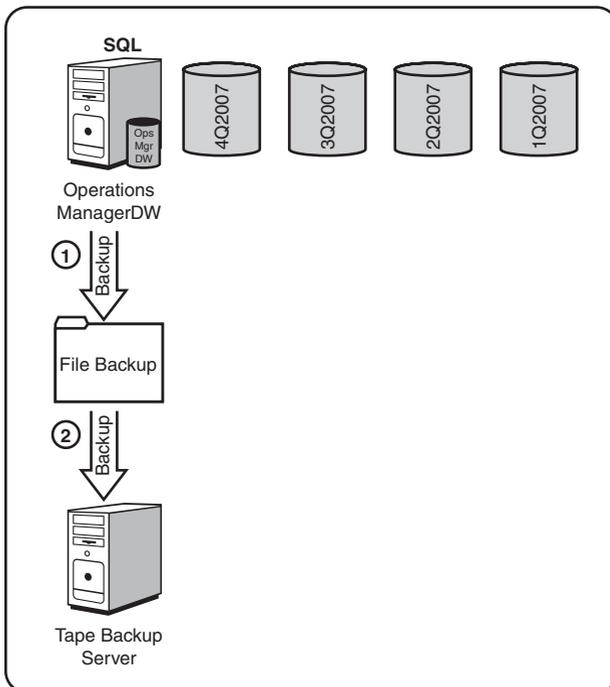


FIGURE 12.2 Backup of data warehouse with archives.

You can perform this backup process weekly, daily, or at whatever period meets your business requirements, with the procedure remaining the same. The amount of disk storage required is based on the size of the databases, which is based on the data captured by the

agents. Calculating size for a 500GB database with one quarter of data, the company will need disk storage to hold five databases (5 x 500GB), plus an additional 500GB for the file backup that is archived to tape. This is a total of 3,000GB (3TB). You can contrast this figure with the original 4TB-plus storage requirement and see we have also conserved on disk storage! We also only need to back up 500GB at a time, rather than 2TB; making the backup operation more efficient.

Procedures are a bit more complex for the end-of-quarter backup process, shown in Figure 12.3. The following steps outline the process of transitioning at the end of 1Q2008:

1. First, the data warehouse is backed up to a disk file. This is an online SQL Server backup, so there is no interruption in availability.
2. Next, the backup file is copied to tape. In the event of a disaster, this tape backup can easily be restored.
3. The backup file is restored to a new SQL Server database storing data for that quarter (in this example, 1Q2008).
4. The database now outside the one-year data retention requirement (1Q2007) is deleted.
5. The tape backup of the data warehouse in Step 2 (1Q2008) is replicated for long-term tape storage.

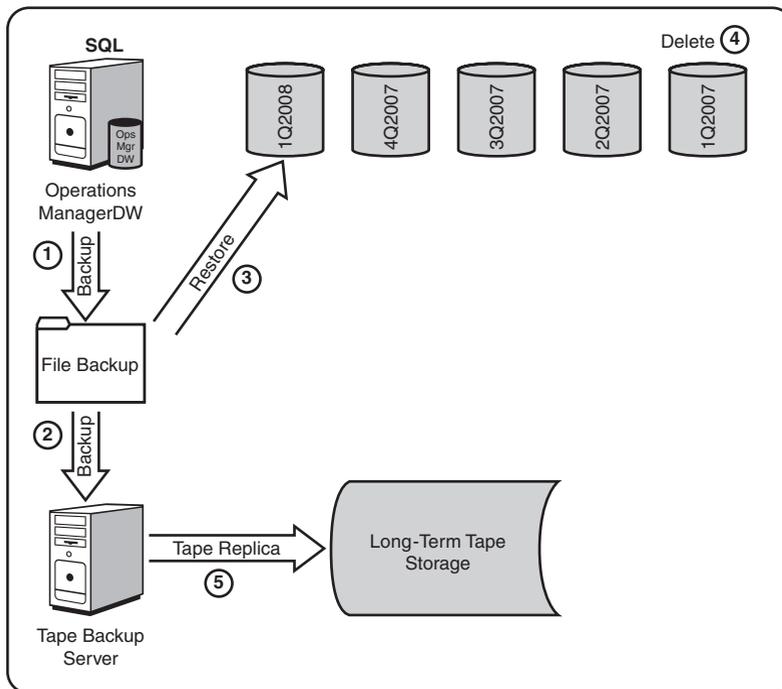


FIGURE 12.3 Quarterly backup of reporting database with archives.

These processes can be automated scripts and jobs, or you can run them manually as end-of-period procedures. The overall process is flexible and can be adjusted to support monthly archives rather than quarterly. The advantage of using monthly archives is that the backups are correspondingly shorter, but the report horizon will be shorter and only cover a single month. You could also extend this process to occur every 6 or 12 months! If it becomes necessary to query data in an archive, you could restore the archived backup as a database file and change the data source in SQL Reporting Services to point to that specific database.

This process is outside of any mechanisms designed or supported by Microsoft.

Grooming the ACS Database

Data is groomed out of the ACS database based on the data retention period specified during setup, with the default being 14 days. The ACS collector calls a SQL procedure to remove a partition that is outside of the data retention period. This procedure can be found on disk at `%SystemRoot%\system32\Security\AdtServer\DbDeletePartition.sql`. The data retention period itself is specified in the `dtConfig` table in the `OperationsManagerAC` database.

To update the data retention period, run the following SQL query:

```
USE OperationsManagerAC
Update dtConfig
SET Value = <number of days to retain data + 1>
WHERE Id = 6
```

To retain 7 days of data, set `<Value> = 8`. Data is accumulated at approximately 7.6MB per day per workstation.

Further ACS database sizing information is available in Chapter 15.

Backing Up and Restoring the SQL Server Databases

Many Information Technology (IT) organizations have a database support group responsible for their Microsoft SQL database servers and in charge of backing up and restoring SQL Server databases. You should work with your database group to ensure that an appropriate backup and restore plan exists for your OpsMgr databases. If you do not have a group responsible for database backups, you will need to create your own backup and restore plan. This plan includes scheduling times for backups, identifying actual database files, and defining procedures for backing up and restoring those files.

Most enterprise backup implementations include a separate software module that can be installed to back up a SQL Server database while it is running. It is highly recommended

that this type of backup agent be employed in your design to provide for online backups of the MOM databases.

Alternatively, you can use SQL Server's backup feature to back up the databases to (local or remote) file or local tape and then back up the resulting files during your normal backup process. This does not require a SQL backup agent and has the advantage of being very fast. The downside is that you need sufficient disk space to hold a backup the size of the entire database, which in the case of the reporting database we know can be quite large.

Database Backups

We will use SQL Server 2005's backup component to back up the Operational database, as an example of the process that you can use for the other databases used by OpsMgr 2007. SQL backups are defined using SQL Server Management Studio. For the Operational database, you should always perform a complete backup—not a differential backup—because by default, Operations Manager supports a simple recovery from a full backup only, not a forward recovery with the transaction log. By default, members of the sysadmin fixed server role and the db_owner and db_backupoperator fixed database roles have permission to back up a SQL Server database.

TIP

Types of Database Recoveries

Without getting too much into database technology, Microsoft SQL Server supports three types of recovery—full, bulk_logged, and simple. A *full* recovery uses the database and transaction log backups for full recoverability. *Bulk_logged* uses minimal transaction logging for bulk load types of operations—if a recovery is necessary, those transactions must be reapplied.

Simple, which is used by the OperationsManager, OperationsManagerDW, and OperationsManagerAC databases, recovers the database without using the transaction logs as part of the process. Using a simple recovery model, any database changes made since the previous full backup are lost. However, this model is the simplest to perform restore operations on.

(This is not to say you cannot change the recovery mode. Chapter 10, “Complex Configurations,” discusses the procedures to set the Operational database to Full Recovery and configure log shipping. You can use log shipping on the Data Warehouse database as well. You may decide to implement log shipping for a high availability environment.)

Backup Steps

You should back up the Operational, Data Warehouse, and ACS databases daily. The following procedure defines a backup job for the Operational database:

1. In the left panel of SQL Server Management Studio, navigate to Databases -> OperationsManager. Right-click on the OperationsManager database, select Tasks, and then choose Backup, which brings up the Back Up Database General page shown in Figure 12.4.
2. The default backup type is Full, which is the backup type we will use for the OperationsManager database. A Full database backup backs up the entire database file, rather than just those changes since the previous backup.

Under Destination, select the backup destination, which can be a disk file or tape. Here we will back up the OperationsManager database to disk, which is the default.

3. Select Add under Destination. The wizard provides a default location of *%ProgramFiles%\Microsoft SQL Server\MSSQL.1\MSSQL\Backup*. To enter the location and filename where the backup will be stored, click on the button with the ellipses (...) and name the file OperationsManager_Full_Group1.bak. In this case, we are using a folder named C:\Backups, rather than the default folder. Figure 12.5 shows the location specified for the backup file.

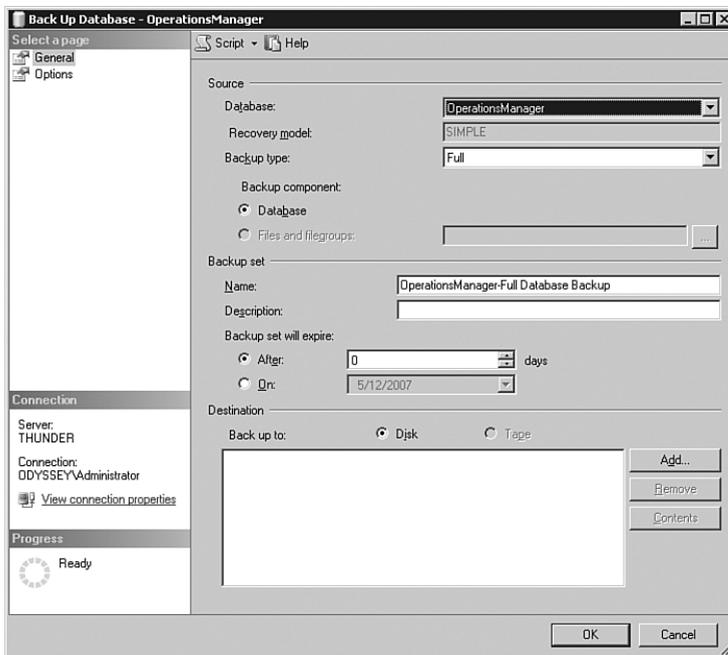


FIGURE 12.4 The SQL Server Management Studio Manager backup screen.

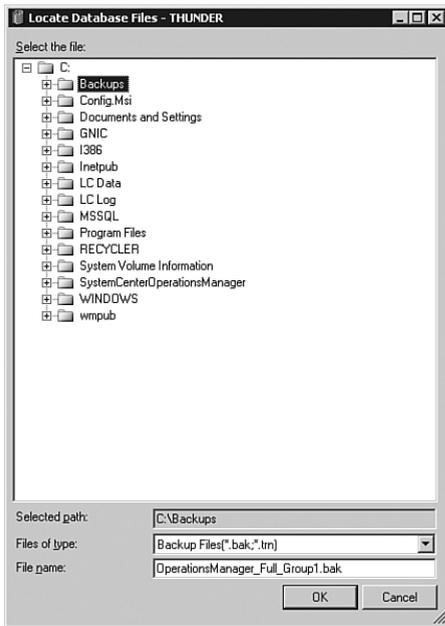


FIGURE 12.5 The backup device location screen.

TIP

Backup Naming Conventions

Because you may have more than one management group, you may want to include the management group name as part of the filename for your backup files. You may also want to include the type of backup, so that a name for the Operational database backup might be `Operational_Full_<management group name>.bak`.

4. SQL Server Management Studio next displays the Select Backup Destination screen, shown in Figure 12.6.
5. After specifying the general requirements for the backup, it is time to move to the Options page shown in Figure 12.7. You must decide whether you will overwrite the file (backup set). By default, SQL Server appends the current backup to the end of the backup file if it already exists. Alternatively, you can overwrite (replace) the file. The option to truncate the transaction log is grayed out because the SQL Management Studio will not let you truncate the log when the database recovery type is defined as simple. If you want to truncate the log, you will have to add a step to do this manually.

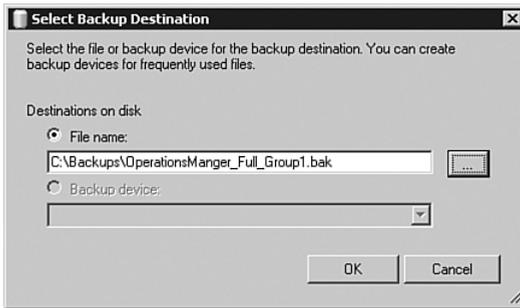


FIGURE 12.6 The Select Backup Destination screen.

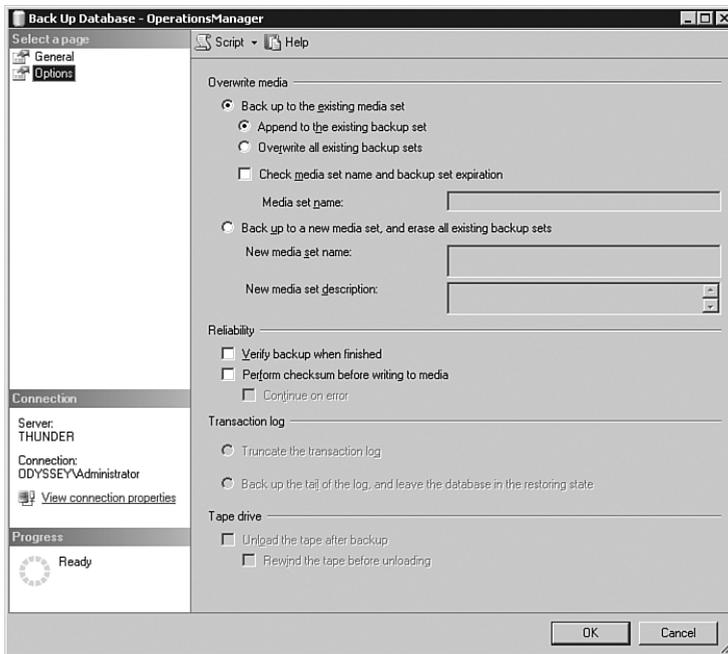


FIGURE 12.7 The backup options page.

Simple recovery is similar to truncate log on checkpoint—meaning in theory you do not need to truncate the log. In earlier versions of SQL Server, the truncate on checkpoint did not always work as advertised as part of the backup process, and the log file would eventually fill up. Some Database Administrators (DBAs) will still add a step to manually truncate the log as insurance.

6. Selecting the Script option at the top of Figure 12.7 generates Transact SQL code you can use to schedule the backup rather than having to return to SQL Management Studio each time you want to back up the database. (After the script is generated, the progress status shows that scripting completed successfully.)
7. After generating the script, select the Script menu at the top of the panel to bring up scripting options. You can select one of several listed options:
 - ▶ Script Action to New Query Window
 - ▶ Script Action to File
 - ▶ Script Action to Clipboard
 - ▶ Script Action to Job

To schedule the backup as a SQL job, select the Script Action to Job option, displayed in Figure 12.8. SQL jobs are run by the SQL Server Agent service.

8. Define the parameters of the backup job. Selecting the Script Action to Job option opens the New Job dialog. At the General page, you can change the owner and category of the job. Figure 12.9 shows the default options.
9. Select the Schedules page (see Figure 12.10) and click New to add a new schedule.

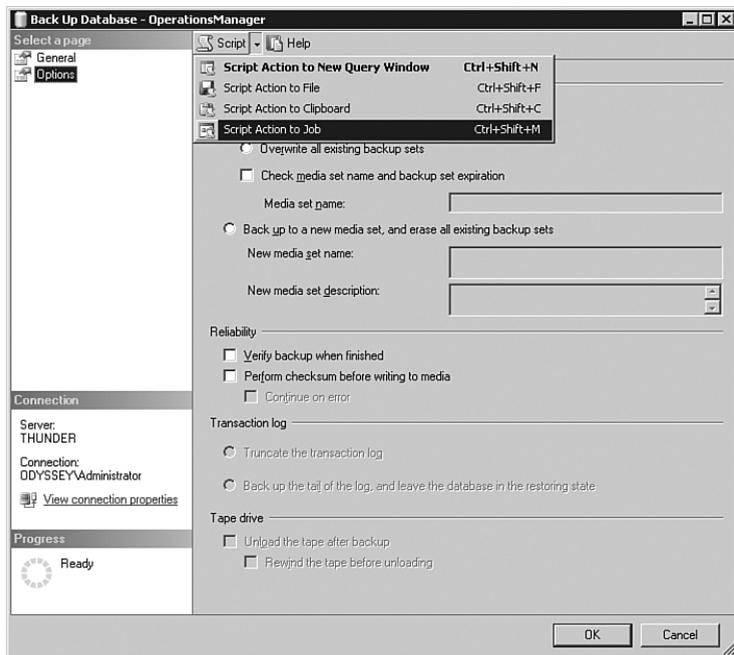


FIGURE 12.8 Create a SQL backup job.

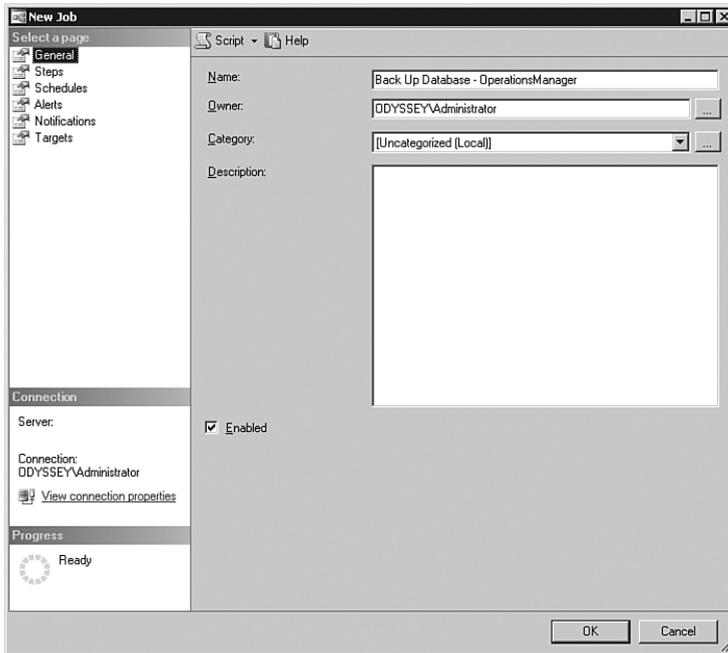


FIGURE 12.9 The New Job screen.

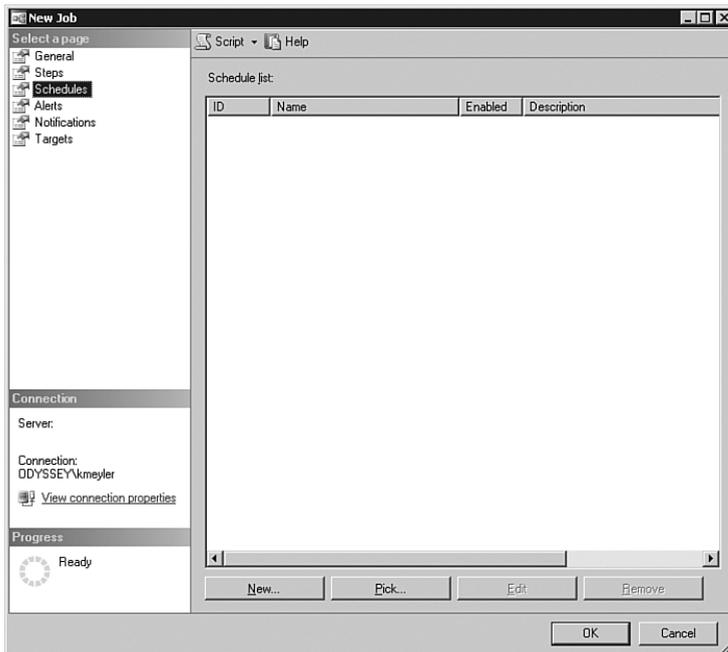


FIGURE 12.10 The New Job schedules page.

10. You can now define the details of the schedule. Figure 12.11 shows a Schedule type of Recurring with a backup frequency of daily and a start time of 3:00 AM. After completing this screen, you can also specify Notifications and Targets as part of the job properties. Click OK to save the job. The job information is saved in the Msdb database.

TIP

Scheduling Database Backups

SQL Server uses an online backup process, allowing a database backup while the database is still in use. During a backup, most operations are possible such as INSERT, UPDATE, or DELETE statements. For general processing efficiency, we recommend you perform backups during a period when you do not expect a high level of updates on the database you are backing up.

FIGURE 12.11 The new job scheduled to recur on a daily basis.

Truncating the Transaction Log

An optional step after creating the database backup step is also to truncate the transaction log.

More on Database Maintenance and Backups

Besides the fairly obvious reason for doing backups mentioned earlier—that of having a database to restore in the event of damage to the database or disk—another reason is to keep the size of the transaction log manageable, which keeps OpsMgr functional.

What is a transaction log? All updates to a SQL Server database are first written to its transaction log. The transaction log exists because SQL Server supports transaction processing. A *transaction* is a logical unit of work—all operations defined as a transaction will either succeed or fail together. For example, assume you want to move \$500 from your checking account to your savings account. If the money is removed from your checking account but never deposited into savings, you have lost \$500—it just disappeared! Transaction processing allows these two operations to be grouped into a single transaction, maintaining data integrity. If your deposit doesn't get to your savings account, the transaction is not "committed"—it is incomplete, and the update to your checking account is "rolled back"—and the \$500 is still in your checking account.

The transaction log keeps track of every data modification performed in your database, who performed it, and when. However, if records are not eventually deleted from the transaction log, the file will fill up—or if autogrow is enabled, the file grows until it fills all available space on the disk holding the physical log files. SQL Server automatically truncates the log every time a checkpoint is processed, which occurs by executing a CHECKPOINT statement, modifying the database using the ALTER command or with SQL Management Studio, or shutting down the SQL Server instance.

Because the Operations Manager databases use a simple recovery model by default—which does not utilize the transaction log during database restores—all log records other than those at the active portion of the log can be truncated at any time, other than when running a database backup.

For those databases using the simple recovery model, SQL Server generates an automatic checkpoint. This automatic checkpoint will delete the inactive portion of the log when the log becomes 70% full (or the number of records in the transaction log reaches the number the SQL Server Database Engine estimates it can process during the time specified by the recovery interval option, whichever occurs first). You can also set a checkpoint yourself in a query to delete the inactive portion of the log using the following syntax:

```
USE OperationsManager
CHECKPOINT
```

If by some change your transaction log should fill up, don't panic. Run the following statement in a SQL query window to remove the inactive portion of the log and not log the operation:

```
DUMP TRANSACTION OperationsManager WITH NO_LOG
```

These commands can also be run against the Data Warehouse and Auditing databases.

The Data Warehouse database has the autogrow option on by default for the data portion of the database. Do *not* set the other databases to autogrow.

Earlier versions of Microsoft SQL Server required truncating the transaction log manually when a database used the simple recovery model. (There was also a bug in SQL Server 2000 [474749] that prevented automatic checkpoints from always running as expected.) The following SQL statement initiates the truncate operation for the Operational database:

```
BACKUP LOG OperationsManager WITH TRUNCATE_ONLY
```

You can add this statement as a job step to the backup job you just created for the database segment. In SQL Server Management Studio, navigate to your database server, select SQL Server Agent -> Jobs, and then edit the OperationsManager backup job, adding a second step with the options shown in Figure 12.12. Be sure to change the Database drop-down box on the General page to specify the OperationsManager database.

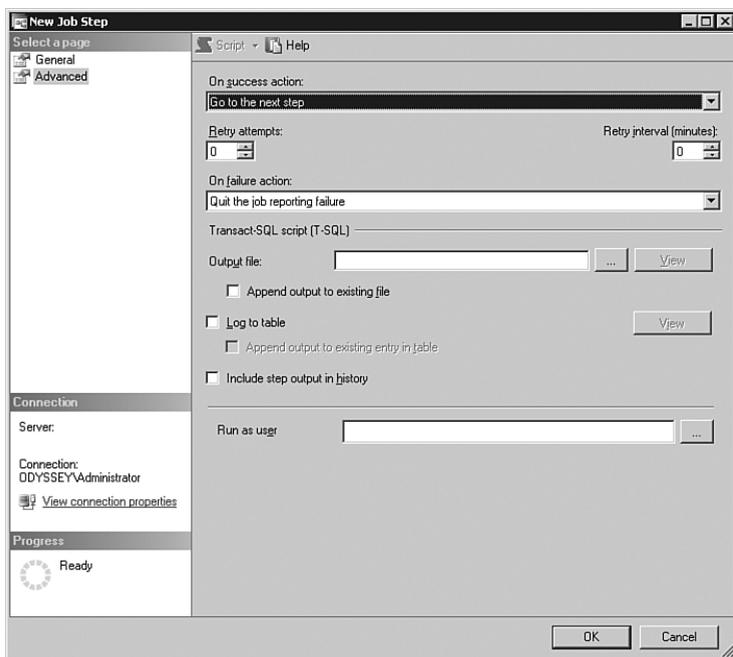


FIGURE 12.12 Truncate the transaction log as part of the backup process.

When you save the job step, you may be asked if you want to change the success action of the last step from “Go to Next Step” to “Quit with Success”. Accept the changes, but then select Step 1 and change the properties of the Advanced page to change the success action to Go to the next step (see Figure 12.13).

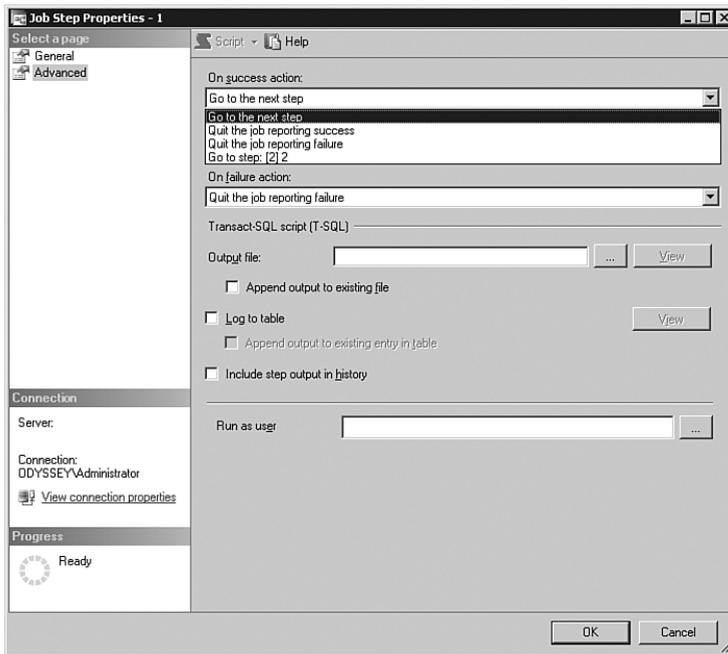


FIGURE 12.13 Change the success action for the first job step.

NOTE

Truncating the Transaction Log in SQL Server 2005

With SQL Server 2005, Microsoft is deprecating the BACKUP LOG WITH TRUNCATE_ONLY functionality, as the log is automatically truncated with the simple recovery model. Although this statement will still work, it will not be supported in the next version of SQL Server.

After defining the database backup, make sure that there are procedures in place to copy the file you create from the database backup to back up media for archival and restoration requirements. You can use your existing nightly file backup process, adding the OperationsManager backup file to the list of files and folders being backed up. In addition, you should make sure that the other files mentioned earlier—such as .mp and .xml files, encryption key files, and any custom files—are also backed up regularly as part of your daily backup procedure.

Database Restores

If one of the Operations Manager databases becomes corrupt or a hardware issue causes you to lose a database, you will need to restore the affected database(s). We will build on the previous example where we created a backup of the Operational database using SQL Server Management Studio. For our scenario, we will assume that the Operational database is corrupt and cannot be repaired. At this point, our strategy is to restore from the latest backup.

The following procedure discusses the process of restoring the Operational database:

1. Be sure to stop the OpsMgr SDK service on the RMS to ensure that Operations Manager will not try to write data to the database. All database access goes through the SDK service.
2. Before performing a full restore for a SQL Server database, you must delete the existing version of the database. Launch SQL Server Management Studio -> Databases -> OperationsManager. Right-click on the database and select Delete. Uncheck the option to delete backup and restore history information from the database; then click OK to delete the Operational database.
3. Restore the database from the last backup. Right-click on Databases and select Restore Database. In the Source for Restore section, select From Database, and select OperationsManager from the drop-down list. This displays the Restore Database screen, as shown in Figure 12.14.

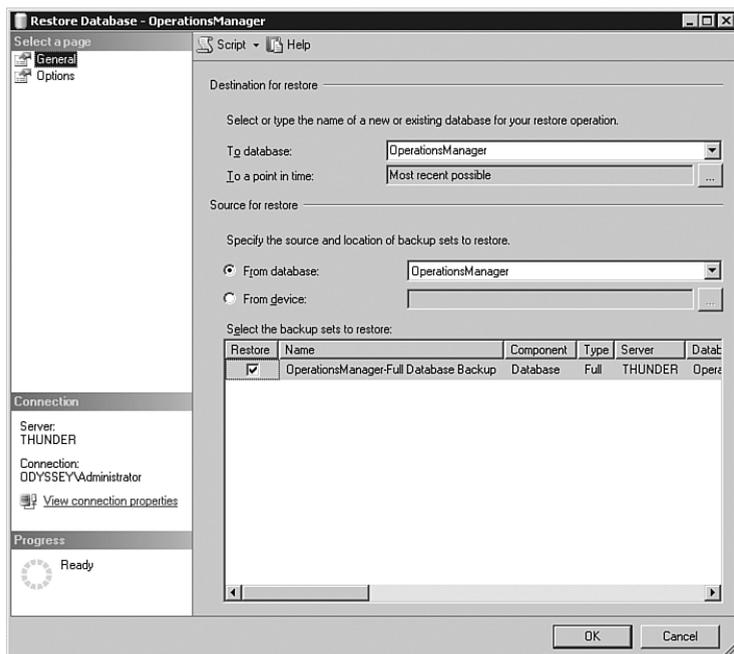


FIGURE 12.14 The Restore Database screen.

4. If you have more than one backup, verify you have selected the latest one for restore and click OK to begin the restore process. Depending on the size of your database, this may take several minutes.

NOTE

Role of OpsMgr Queue Files

Remember that OpsMgr data is stored in queue files when the database is unavailable, which minimizes the chance of loss of current data not yet written to the database.

After you restore the database, restart the OpsMgr SDK service and launch the Operations console. Open the Administration node to verify it has the correct rule groups and configuration. You can also launch the Monitoring node to make sure that agents are sending heartbeats. This ensures that OpsMgr is operational.

TIP

Restoring Databases Used by OpsMgr

When restoring any of your databases, be sure the database server software installed is at the corresponding service level of your database backup. As an example, you cannot install SRS without a service pack and then use a Data Warehouse database that has SQL Server 2005 Service Pack 1 or Service Pack 2 applied to it.

The installation is blocked if you try to install without matching levels of maintenance.

Moving the OpsMgr Databases to a Different Server

As part of a disaster recovery scenario, or perhaps just because it is something that needs to be done, you may have a requirement to move the OpsMgr databases to a different SQL Server database server. The next sections discuss moving the OperationsManager, OperationsManagerDW, and OperationsManagerAC databases.

Moving the Operational Database

The Operational database is installed as part of your OpsMgr setup and cannot be reinstalled without reinstalling the management group. If you need to move the database to another database server, you can perform the following steps:

1. Stop all OpsMgr services on the RMS. If you have multiple management servers, stop the Health service on those machines as well.
2. Using SQL Server Management Studio, connect to the source database server and back up the OperationsManager database.

3. Connect to the destination database server and create Windows/AD SQL logins for the following OpsMgr accounts: SDK, MSA, and DWWA.
4. Copy the OperationsManager database backup file to the destination server, and restore the database to the destination database server.
5. Using SQL Server Management Studio on the destination database server, right-click on the OpsMgr SDK login and go to Properties (Security -> Logins -> <domain>\SDK login name -> Properties).
6. In the properties for the Operations Manager SDK account, go to the User Mapping page and click on the OperationsManager database. Ensure that the following database roles have been assigned to the SDK account:
 - ▶ Db_datareader
 - ▶ Db_datawriter
 - ▶ Db_ddladmin
 - ▶ Db_owner
 - ▶ Dbmodule_users
 - ▶ Sdk_users
7. Click OK.
8. On the RMS and each of your management servers, open REGEDIT. Browse to HKLM\Software\Microsoft\Microsoft Operations Manager\3.0\Setup, and update the string called DatabaseServerName to reflect the name of the new database server.
9. Reboot the RMS and other management servers.
10. Set Enable Broker as follows:
 - ▶ Open SQL Server Management Studio and connect to the database server now running the Operations database.
 - ▶ Open a new Query window and type the following syntax:
ALTER DATABASE OperationsManager SET SINGLE_USER WITH ROLLBACK IMMEDIATE
 - ▶ Execute the query, then enter and execute the following query:
ALTER DATABASE OperationsManager SET ENABLE_BROKER
 - ▶ Close SQL Management Studio and reopen it, and again connect to the database server now running the Operations database.
 - ▶ Open a Query window, and enter and execute the following syntax:
ALTER DATABASE OperationsManager SET MULTI_USER

- ▶ To verify `ENABLE_BROKER` is set to 1, connect to the Master database and run the following query:

```
SELECT is_broker_enabled FROM sys.databases WHERE name=  
➔ 'OperationsManager'
```

11. Restart the OpsMgr SDK and Config services on the RMS, and the OpsMgr Health service on the RMS and all management servers. You may also need to restart the SQL Server and SQL Agent services on the Operations Database Server Component.

While these steps move the Operations database, some configuration data is stored in the `sys.messages` system view in the Master database as part of the SetupOM process. This includes error messages specific to Operations Manager that are not tied to the Operations database.

As part of the CD content for this book, we include a script provided by Matt Goedtel of Microsoft that ensures the error messages specific to the OperationsManager database are available on the new SQL Server database instance. Run this script in the SQL Management Studio, and execute it against the Master database.

TIP

On the CD

Run the SQL script `Fix_OpsMgrDB_ErrorMsgs.SQL.sql` after moving the Operational database to another database server. Be sure to run it against the Master database on the new server.

Moving the Data Warehouse Database

You can take a “simpler approach” for moving the Data Warehouse database than with the Operational database. As OpsMgr Reporting is installed after the management group is created, we can be more straightforward and uninstall the OpsMgr Data Warehouse Component, install it on a different server, and then copy over the original database. However, there are several other changes to make, so it’s not quite as simple as it sounds. Perform the following steps:

1. On the RMS, stop the SDK and Config services.
2. On the RMS and all other management servers, stop the Health Service. Stopping the OpsMgr services prevents updates from being posted to the databases while you are moving the data warehouse.
3. On the current Data Warehouse server, use SQL Management Studio to back up the Data Warehouse database (default name: OperationsManagerDW) to a shared folder on the server. You will want to back up the Master database as well, as a precaution.

4. On the current Data Warehouse server, uninstall the OpsMgr Data Warehouse Component. Open Control Panel -> Add/Remove Programs, select the System Center Operations Manager 2007 Reporting Server, and choose Change. In the Reporting Setup, select Modify, and then select the Data Warehouse Component to not be available. Note that this does not physically remove the data warehouse database as a SQL Server database. After removing the Data Warehouse Component from OpsMgr, delete it manually using SQL Management Studio (assuming you backed it up in step 3!).
5. On the new Data Warehouse server, install the OpsMgr Data Warehouse Component by running OMSetup.exe. Select the option to Install Operations Manager 2007 Reporting, selecting ONLY the Data Warehouse Component for installation. (Mark the Reporting Services Component to not be available on this server, as it is still installed elsewhere.)
6. On the new Data Warehouse server, copy the backup of the Data Warehouse database (step 3) to a local folder. (If the shared folder on the original server is accessible as a mapped drive from SQL Management Studio, you can skip this step.)
7. On the new Data Warehouse server, use SQL Management Studio to restore the data warehouse database backup.
 - ▶ Delete the existing database first; be sure the default option to Delete backup and restore history information for databases is checked.
 - ▶ Restoring the original data warehouse database is necessary to not lose the report data you have already collected for your management group.
8. On the new Data Warehouse server, create a login for the SDK account, the Data Warehouse Action account, and the Data Reader account in SQL Management Studio. Ensure the database permissions are correct for these accounts.
9. On the RMS, start the SDK service.
10. On the server running SQL Reporting Services, modify the data source. In Internet Explorer, open <http://localhost/reports> (add `<$instancename>` to the URL if using a named instance). On the Properties page, choose Show Details. The data source is named "Data Warehouse Main." Select that data source, and in the connection string, change the name of the database server from the old data warehouse server to the new data warehouse server. Click Apply to save your changes.
11. Change the name of the data warehouse server in the OpsMgr databases. Open SQL Server Management Studio to do your edits. For the OperationsManager database, go to the MT_Datawarehouse table and change the value of the MainDatabaseServerName_16781F33_F72D_033C_1DF4_65A2AFF32CA3 column (that really is the column name!) to the new data warehouse database server. For the OperationsManagerDW database, navigate to the MemberDatabase table and change the value of ServerName. Be sure to close the Management Studio when you are through, to save your changes.

12. Restart the Config and Health services on the RMS and the Health service on all management servers.

TIP**The Data Warehouse Server Name Is Stored in the OpsMgr Databases**

In step 10 of the process to move the Data Warehouse database, we change the name of the data warehouse server in both the OperationManager and OperationsManagerDW database. This information is stored as metadata in the databases.

When the management group “wakes up,” it needs to find its data warehouse. It looks in the OperationsManager database to find it, as the information is not stored anywhere in the Registry. However, it is possible that the data warehouse may be partitioned, consisting of multiple databases located on different servers—you may decide to store performance samples on one server, events on another, and so on, to achieve greater scalability.

The data warehouse server metabase information stored in the OperationsManager database can be considered a “master” database for the data warehouse. The system will then query the MemberDatabase table in the “master” itself, to find out where to forward particular types of data from a particular management group.

Moving the Audit Collection Database

Each ACS collector writes to its own database. This database can be installed on the Audit Collector Component or elsewhere. To move the ACS database to another database server, perform the following steps:

1. On the original ACS database server, stop the Audit Collection service. Now use SQL Server Management Studio to backup the database (OperationsManagerAC by default) to a shared folder on the server. (As always, you should also backup the associated Master database.)
2. Using SQL Server Management Studio, delete the OperationsManagerAC database. Be sure that the Delete backup and restore history information for databases and Close existing connections options are both checked.
3. On the new database server, use SQL Server Management Studio to restore the backup. You can either access the database backup from step 1 by first copying the backup file to a local drive, or map a local drive to the shared folder.
4. On the new Audit Collection database server, use SQL Server Management Studio to create a login for the ACS server. The format is `<domain\computername$>`, where *computername* is the name of the ACS server.

5. In SQL Server Management Studio, set the correct permissions for this account in the Security -> Logins folder -> Properties -> User Mapping. Check the box in the Map column that corresponds to the OperationsManagerAC database, and then select db_owner in the Database role Membership for: OperationsManagerAC list.
6. On the computer hosting the Audit Collection service, edit the Registry key HKEY_LOCAL_MACHINE\Software\ODBC\ODBC.INI\OpsMgrAC. Double-click the Server value, and set it to the name of the new ACS database server. Now start the Audit Collection service on this server.

To verify the database move was successful, use SQL Server Management Studio to check the OperationsManagerAC database for entries in the most recent dtEvent_<GUID> table. The datetime stamp should be more recent than when you restarted the Audit Collection service.

TIP

Documenting Your Databases

A useful utility you may want to check out is SqlSpec by ElsaSoft. SqlSpec generates easy-to-read database documentation not only for Microsoft SQL Server but also for many other database platforms. Information is available at <http://www.elsasoft.org/features.htm>, and Stefan Stranger provides several examples of using it against his OperationsManager database at http://weblog.stranger.nl/documenting_your_operationsmanager_database_use_sqlspec.

Backing Up the RMS Encryption Keys

Microsoft provides the SecureStorage Backup tool (SecureStorageBackup.exe) to back up the RMS encryption keys. The syntax is as follows:

```
SecureStorageBackup.exe <Backup|Restore> <BackupFile>
Backup           Backs up encryption keys to file specified as <BackupFile>
Restore          Restores encryption keys stored from <BackupFile>
BackupFile       Specifies file name where the keys will be backed up to and restored
                 ➔from
```

Real World—Always Back Up the Encryption Keys!

If you lose your RMS and have not backed up the encryption keys, you must reinstall the RMS and your management group (Service Pack 1 incorporates some relief for this, see the “Recovering from a RMS Loss” section).

Also, be aware that after you upgrade your OpsMgr infrastructure to Service Pack 1, your previous RMS key backups will not work. An organization in Texas testing the service pack was bit by this when their RMS failed and the recovery did not work since the keys were backed up from prior to SP 1!

Microsoft plans to make it easier to backup the RMS key after your SP 1 upgrade by including a step in the upgrade wizard prompting you to do the backup at the end of the upgrade.

The SecureBackupStorage utility is located on the Operations Manager installation media in the \SupportTools folder and should be copied to the Operations Manager installation folder (%ProgramFiles%\System Center Operations Manager 2007).

The following procedure backs up the encryption key:

1. Log on to the RMS using an account that is a member of the Administrators group.
2. Select Start -> Run ->; then type **cmd**, and click OK.
3. At the command prompt, navigate to %ProgramFiles%\System Center Operations Manager 2007. The utility must be run from the OpsMgr installation directory. Remember, you must first copy this file from the installation media.

NOTE

Directory for Running the SecureStorageBackup Utility

If you do not run SecureStorageBackup.exe from the OpsMgr installation directory, you will get errors about dlls that are not registered.

4. Back up the encryption keys by typing the following:
SecureStorageBackup Backup c:\backups\BackupKey.bin
5. You are prompted to enter a password (twice to confirm). This password is used for storage/retrieval, and must be at least eight characters.
6. The encryption key is saved to the folder and file you specify (c:\backups\BackupKey.bin, in this example). Be sure to remember the retrieval password!

To restore the encryption keys, open a command prompt and navigate to the Operations Manager installation folder (`%ProgramFiles%\System Center Operations Manager 2007`), and execute `SecureStorageBackup Restore <BackupFile>`. You will be prompted to enter the retrieval password. Using the backup key file we created, the syntax for the restore command would be as follows:

```
SecureStorageBackup Restore c:\backups\BackupKey.bin
```

You can also use the `SecureStorageBackup` utility to move the RMS capability to another management server, which we discuss in the next section.

Recovering from a RMS Loss

The RMS has a unique role in an OpsMgr environment. Although you can have multiple management servers accepting data from agents, only the RMS communicates directly with the OpsMgr databases. Given the importance of this role, it is not only important to back up the RMS encryption keys (see the “Backing Up the RMS Encryption Keys section”), but also to be able to transfer the RMS role to another management server if this server will be unavailable for a period of time. This section discusses the steps to restore the RMS role to another management server, as follows:

1. Confirm you have a working RMS and second management server. Figure 12.15 shows our RMS (Hydra) and a management server (DeathSting) in the Operations console.
2. Copy the `SecureStorageBackup.exe` and `ManagementServerConfigTool.exe` utilities to the Operations Manager installation folder on the RMS (`%ProgramFiles%\System Center Operations Manager 2007`). These files are available on the Operations Manager installation media in the `\SupportTools` folder. For our environment, the RMS is Hydra.
3. Run the `SecureStorageBackup.exe` tool, exporting the encryption keys file to a file share. The tool is run by opening a command prompt (Start -> Run -> and then type **cmd**), navigating to `%ProgramFiles%\System Center Operations Manager 2007`, and typing the following command:

```
SecureStorageBackup Backup <BackupFile>
```

where `<BackupFile>` is the shared path and filename of the backed up encryption key.

4. You are prompted to enter a password (twice to confirm). This password is used for storage/retrieval, and must be at least eight characters.
5. Be sure that the keys file is on a file share accessible from the other management server (DeathSting).
6. Copy the `SecureStorageBackup.exe` and `ManagementServerConfigTool.exe` utilities to the Operations Manager installation folder on the other management server (`%ProgramFiles%\System Center Operations Manager 2007`). These files are available on the Operations Manager installation media in the `\SupportTools` folder.

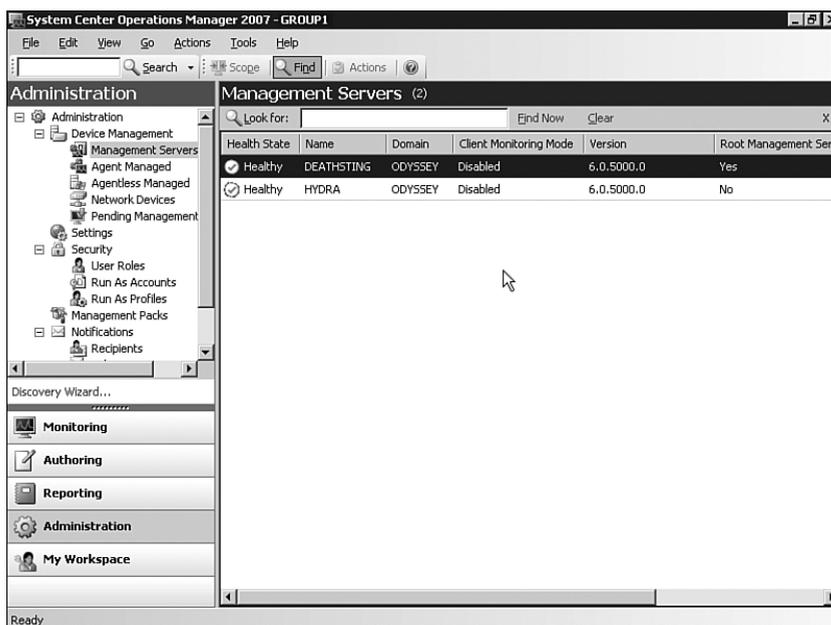


FIGURE 12.15 Initial Root Management Server and Management Server before changing roles.

- From the command prompt in the %ProgramFiles%\System Center Operations Manager 2007 folder, run the SecureStorageBackup.exe tool to restore the key, using the following syntax:

```
SecureStorageBackup Restore <BackupFile>
```

where <BackupFile> is the shared path and filename of the previously backed up encryption key. Enter the password you entered when you created the keyfile.

- At the command prompt, run the ManagementServerConfigTool.exe utility to promote the management server:

```
ManagementServerConfigTool.exe PromoteRMS /DeleteExistingRMS:true
```

You will receive a warning message:

```
Running this tool can cause irreversible damage to
your Operations Database.
Type Y to continue to promote the Management Server
to become the Root Management Server.
```

- Type Y (yes) to continue. The utility completes and displays the information in Figure 12.16.

```

C:\Program Files\System Center Operations Manager 2007>
1 PromoteRMS
Running this tool can cause irreversible damage to your Operations Manager DB. Please
backup your DB before continuing. Continue the PromoteRMS action? (Y/N)
y
Stopping HEALTHSERVICE service on DEATHSTING..stopped.
Updating local settings on DEATHSTING for promotion.
Warning: DEATHSTING is already marked as an RMS!
Warning: DEATHSTING does not have a parent health service
Starting OMSDK service on DEATHSTING..started
Stopping OMSDK service on Hydra.Odyssey.com..stopped.
Stopping OMCFCG service on Hydra.Odyssey.com..stopped.
Stopping HEALTHSERVICE service on Hydra.Odyssey.com..stopped.
Updating class structure to reflect changes:
Promoting DeathSting.Odyssey.com
Demoting Hydra.Odyssey.com
Please be patient this may take some time.
Starting HEALTHSERVICE service on DEATHSTING.....started
Starting OMCFCG service on DEATHSTING..started
Updating local settings on Hydra.Odyssey.com for demotion.
Warning: the OMCFCG service on Hydra.Odyssey.com was set to Auto
Warning: the OMSDK service on Hydra.Odyssey.com was set to Auto
Starting HEALTHSERVICE service on Hydra.Odyssey.com...started
PromoteRMS performed successfully

```

FIGURE 12.16 Changing the role of the Root Management Server to a management server.

10. Restart the Health Service on the original RMS. From the command prompt window in step 3, type the following commands:

```

Net Stop OpsMgr Health Service
Net Start OpsMgr Health Service

```

11. On the newly promoted RMS, open the Operations console. You are prompted for the name of the new Root Management Server to connect to.

Figure 12.17 shows the server roles reversed. The original RMS server is now a management server and the management server is now the RMS.

The full syntax for the ManagementServerConfigTool is included in Chapter 10.

TIP

Creating a New RMS Encryption Key with SP 1

With OpsMgr 2007 Service Pack 1, Microsoft includes a CREATE_NEWKEY command line switch that allows you to recover a lost RMS without a backed up encryption key. The switch is used when running MOM.msi on the management server that will become the new RMS.

J.C. Hornbeck of Microsoft provides preliminary documentation about this switch at <http://blogs.technet.com/smsandmom/archive/2007/12/05/opsmgr-2007-what-if-i-lose-my-rms-encryption-key.aspx>. You should still backup your encryption key, but this new capability gives you a way to recover without having to rebuild.

Other Components to Update After Moving the RMS

When you move the RMS to another management server, you will also need to update the Reporting Server and the Web Console Server with the new location of the RMS.

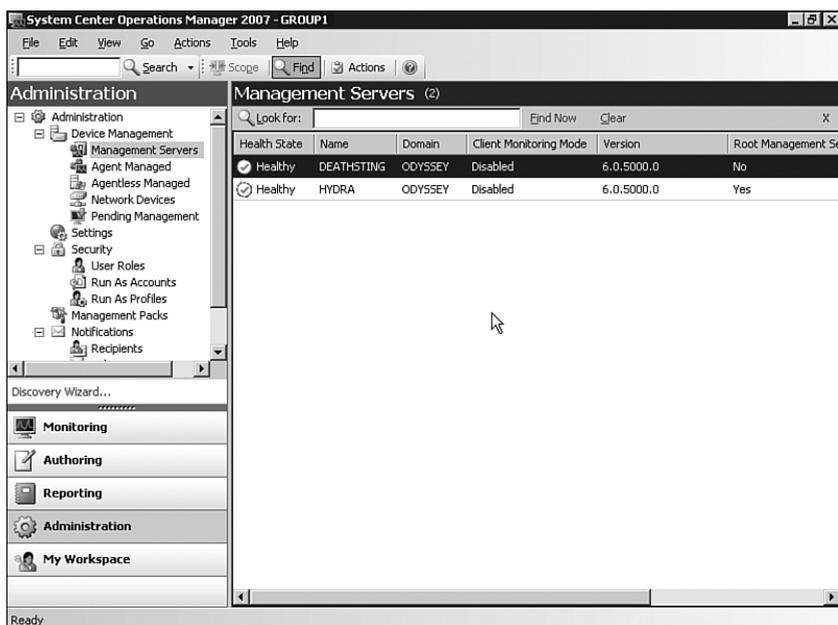


FIGURE 12.17 The Root Management Server and the management server after changing roles.

Perform the following steps on the Reporting Server:

1. On the Reporting Server, navigate to `%ProgramFiles%\Microsoft SQL Server\MSSQL.2\Reporting Services\ReportServer`.
2. Open the `rsreportserver.config` file using Notepad.
3. Find the two entries for `<ServerName>` and change it to the new RMS name.

Now perform the following steps on the Web Console Server:

1. On the Web Console Server, navigate to `%ProgramFiles%\System Center Operations Manager 2007\Web Console`.
2. Open the `Web.config` file using Notepad.
3. In the `<configuration>` section, find the following:


```
<!--This is internal connection between the web server and the MOM server .-->
<add key= "MOMServer" value="Hydra.Odyssey.com" />
```
4. Change the contents of **value** from the old RMS name (using the Fully Qualified Domain Name) to the new RMS name (specify the Fully Qualified Domain Name)—for example, `value= "DeathSting.Odyssey.com"`.

See KB article 555950 for additional information, at <http://support.microsoft.com/kb/555950>.

The `rsreportserver.config` and `web.config` files will now contact the new RMS.

For pre-SP 1 OpsMgr 2007 environments, Microsoft confirms there are additional issues with promoting a management server to the RMS role, as the data warehouse processing is still on the old RMS after the promotion. The data warehouse operations code and promotion code have a “misunderstanding” such that the data warehouse operations are not moved to the new RMS. The synchronization process assumes the SDK is local, but it actually is not—as the RMS has moved and the SDK service is stopped on the old RMS. (The SDK service moves management pack information between the operational and data warehouse databases.)

There is no “easy” fix except for starting the SDK service on the old RMS, which takes care of the data transfer. Once SP 1 is in place, you can promote some other management server to be the RMS and then back to move your data warehouse processing to the real RMS.

Restoring a Clustered RMS

If your RMS is on a cluster, the disaster recovery process is a bit more interesting, as you will be reinstalling the RMS and the Operational database. The high-level recovery steps are as follows:

1. Back up the Operations Manager database to a separate system (disk or tape). See the “Database Backups” section of this chapter for specific steps.
2. Back up the RMS Encryption key, which we describe in the “Backing Up the RMS Encryption Keys” section.
3. Create a new clustered RMS configuration in the same fashion as the previous management group:
 - ▶ Reinstall the Operations Database Server Component using the same management group name as was originally used.
 - ▶ Reinstall management servers on all cluster nodes.
 - ▶ Back up the new encryption key from the new RMS (this is the first cluster node on which a management server was installed), in order to create the new clustered RMS.
 - ▶ Restore the new encryption key on all the other cluster nodes.
 - ▶ Run the `ManagementServerConfig.exe` tool using the `InstallCluster` action (this tool is documented in Chapter 10).

Specifying `InstallCluster` removes all the local changes on the cluster nodes that will compromise its recovery. Running this tool requires that all OpsMgr services are stopped on all management servers.

4. Drop the new Operational database and restore the original database (see the “Database Restores” section for additional information).
5. Restore the original encryption key on all cluster nodes using the SecureStorageBackup.exe tool.
6. Using the Cluster Administrator, bring the clustered RMS back online.
7. In SQL Server Management Studio, run the following query:

```
SELECT is_broker_enabled FROM sys.databases WHERE  
name='OperationsManager'
```

If the returned value is “0,” you will need to reset the broker service, using these SQL queries:

```
ALTER DATABASE OperationsManager SET SINGLE_USER WITH ROLLBACK IMMEDIATE  
ALTER DATABASE OperationsManager SET ENABLE_BROKER
```

Close SQL Management Studio and reopen it; then run this query:

```
ALTER DATABASE OperationsManager SET MULTI_USER
```

8. Restart the SQL services if they are stopped; then restart the SDK service.

Backing Up Management Packs

It is just as important to back up modifications to management packs as it is to back up the SQL Server databases used by Operations Manager. All overrides and custom rules in monitors are saved to a user-specified unsealed management pack stored in the Operational database. Backing up and maintaining management packs as separate objects gives you the granularity to restore a specific management pack rather than the entire database. You may want to create a separate management pack to store your overrides for each of the various management packs. Microsoft provides an unsealed management pack, named *Default*, where you may also be storing changes (although this is typically not recommended).

You can back up (export) unsealed management packs in an ad-hoc manner using the Operations console. We discuss this technique in Chapter 13, “Administering Management Packs.” For purposes of regularly scheduled backup jobs, we suggest that you back up your unsealed management packs in a batch mode, using PowerShell cmdlets to export the management packs from the RMS.

NOTE

Implementing a Full Change Control Process

A good change control process will include backing up management packs and storing them in a code repository solution.

Microsoft's PowerShell command-line shell is a new scripting language, allowing IT professionals to more easily control system administration. PowerShell is included with Windows Server 2008, Exchange Server 2007, and Operations Manager 2007. The Operations Manager Command Shell is a customized instance of PowerShell; it is a superset of PowerShell with cmdlets specific to OpsMgr functions. You can make OpsMgr SDK calls using the Command Shell.

Chapter 8, "Configuring and Using Operations Manager 2007," included a PowerShell example of importing management packs. In this chapter, we will export management packs using the `export-managementpack` PowerShell cmdlet.

Microsoft helps make the cmdlet easy to use by giving us a syntax example when you type **get-help** for this cmdlet. The example will export all unsealed management packs (`$.Sealed -eq $false`) to the root of the C:\ drive:

```
$mps = get-managementpack | where-object {$.Sealed -eq $false}
foreach($mp in $mps)
{
export-managementpack -managementpack $mp -path "C:\"
}
```

To put this syntax in a script, perform the following steps:

1. Open Windows Notepad and add code that initiates a connection to the RMS and loads the OpsMgr PowerShell extensions:

```
param ($ServerName)
add-pssnapin
"Microsoft.EnterpriseManagement.OperationsManager.Client";
set-location "OperationsManagerMonitoring::";
new-managementGroupConnection -ConnectionString:$ServerName;
set-location $ServerName;
```

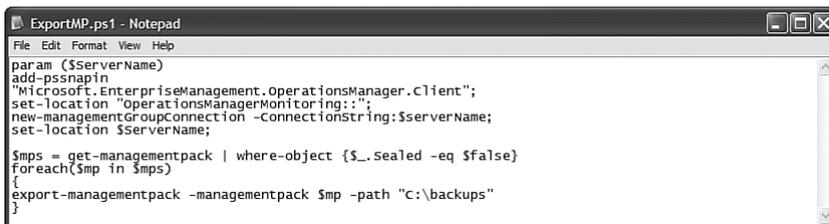
Because we are loading the OpsMgr extensions, we can run our script from the standard PowerShell environment, without having to run from inside the OpsMgr Command Shell. This is useful if you want to run scripts in a batch mode.

2. Now, add the original code from our `get-help` example, modified to export the management packs to C:\backups:

```
$mps = get-managementpack | where-object {$.Sealed -eq $false}
foreach($mp in $mps)
{
export-managementpack -managementpack $mp -path"C:\backups"
}
```

3. Save the text file with a `.ps1` extension for it to be executable by PowerShell. Figure 12.18 shows Notepad open to the `ExportMP.ps1` PowerShell script we created to export all unsealed management packs.

To run the script shown in Figure 12.18 (saved as Z:\ExportMP.ps1), perform the following commands: Start -> Run > CMD, then type **PowerShell**. This opens a PowerShell instance.



```

param ($ServerName)
add-psnapin "Microsoft.EnterpriseManagement.OperationsManager.Client";
set-location "OperationsManagerMonitoring:";
new-managementGroupConnection -ConnectionString:$ServerName;
set-location $ServerName;

$mps = get-managementpack | where-object {$_.sealed -eq $false}
foreach($mp in $mps)
{
export-managementpack -managementpack $mp -path "C:\backups"
}

```

FIGURE 12.18 A PowerShell script to export all unsealed management packs.

From the PowerShell prompt, type the following, specifying the RMS for the ServerName variable:

```
Z:\ExportMP.ps1 -ServerName:<rootmgmtserver>
```

How the ExportMP PowerShell Script Works

The script looks for the SDK service, which runs on the RMS only. If the RMS is not properly specified as the `-ServerName` value, you will receive the following error:

```
New-ManagementGroupConnection : The sdk service is either not running or not yet
initialized.
```

If you attempt to run the script from the same directory it is located, be sure to prepend the script name with a `“.\”`. For this particular example, the script name would be typed `.\ExportMP.ps1`.

Another Approach for Exporting Management Packs

If you want to back up management packs associated with a certain application or service, you can use the following PowerShell Script as a starting point. This example, which also uses the `export-managementpack` cmdlet, searches for management pack names that include the literal `“AD”` and then exports all management packs (sealed or unsealed) related to Active Directory:

```

param ($ServerName)
add-psnapin "Microsoft.EnterpriseManagement.OperationsManager.Client";
set-location "OperationsManagerMonitoring:";
new-managementGroupConnection -ConnectionString:$ServerName;

```

```

set-location $ServerName;
$ mps = Get-ManagementPack | where{$_Name -match 'AD'}
=foreach($mp in $ad_mps)
{
export-managementpack -ManagementPack:$mp -Path:"c:\backups"
}

```

When you run this script, you will notice that all exported management packs have an extension of .xml—even the sealed ones. This is by design. Microsoft created sealed management packs (MPs) so you will not edit directly into the management pack; the vast majority of support calls for MOM 2005 were from problems with customers changing something inside of a management pack. With sealed MPs, you must use overrides to change rules and monitors, and those overrides are stored separately from the sealed MP.

Save the preceding code to a file with a .ps1 extension, open a PowerShell instance: Start -> Run > and type CMD, and then type **PowerShell**. Run the script by typing the following command:

```
Z:\mpexportAD.ps1 -ServerName:<rootmgmtserver>
```

You can build on this example to develop other customized export scripts.

Backing Up Management Packs from a Batch File

Now, let's take it one more step. We can create a batch file that invokes a PowerShell script, which can be part of a nightly backup routine. We will save our ExportMP.ps1 script to our c:\backups directory, and execute it from a batch file containing the following command:

```
%systemroot%\system32\windowspowershell\v1.0\powershell.exe
➔c:\Backups\exportMP.ps1 -ServerName:<rootmanagementserver>
```

The command invokes the PowerShell environment and runs our PowerShell export script. Running this nightly ensures that all customized management packs are extracted to the file system on a regular basis, where they can then be backed up to tape as part of your standard backup process.

Using the Operations Console

Chapter 13 discusses exporting and importing management packs. The functionality can also be used as part of your backup strategy. This chapter focuses on those steps of particular significance when backing up management packs.

The Operations console only allows you to export unsealed management packs. To use the Operations console to back up and restore management packs, perform the following steps:

1. In the Administration node of the Operations console, right-click on the specified management pack in the Details pane to bring up the option to Export Management

Pack, shown in Figure 12.19 (for sealed management packs, this option is grayed-out). This option is also available in the Actions pane, although it is grayed-out unless you have highlighted an unsealed management pack.

We will back up (export) the Default management pack, which is where all management pack changes are saved, by default.

2. You are asked to browse to the folder where the exported management pack will be saved. Figure 12.20 shows that we will export the Default management pack to the C:\backups folder. We are not given the option to specify a filename.

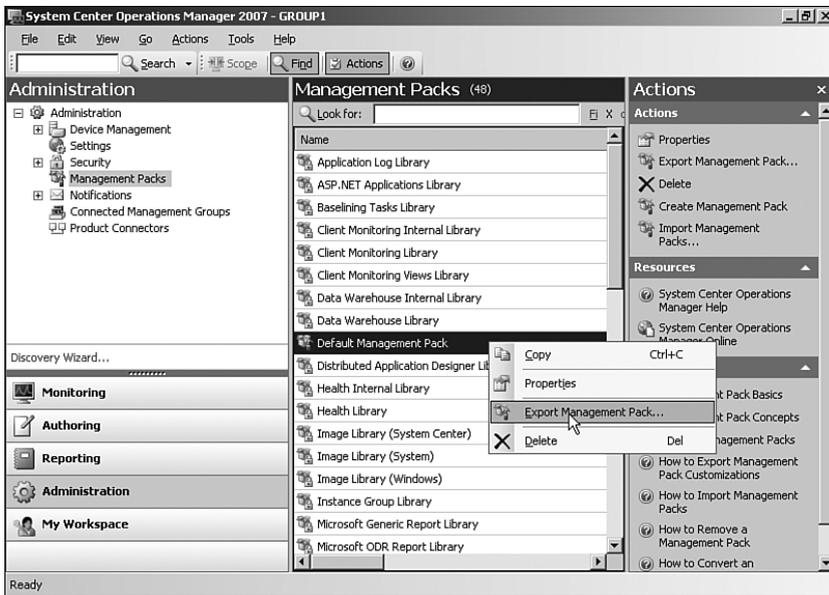


FIGURE 12.19 Specifying a management pack to export.



FIGURE 12.20 Specifying the folder to store the management pack.

3. If the management pack has already been exported to that folder, you are asked if you want to overwrite it (see Figure 12.21). If you choose No, the management pack is not exported and you are not given an option to save to another filename or folder.
4. Successfully exporting the management pack brings up the message displayed in Figure 12.22.

Restoring the management pack using the Operations console is accomplished using the Import function discussed in Chapter 13.

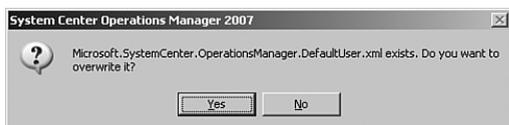


FIGURE 12.21 Prompt to replace previously exported management pack.



FIGURE 12.22 Management pack export complete.

Backing Up Reports

Before we discuss backing up reports, let's talk about how reports work. Similar to MOM 2005, Operations Manager 2007 uses SQL Reporting Services as its report engine, although the Operations console now front-ends and displays the reports. You can also access the OpsMgr reports from the SQL Reporting console, but that is not the recommended approach, as the report names display with their GUIDs (Globally Unique Identifiers) and the information (metadata) about the report is only displayed in the Operations console.

When you display the report hierarchy in the Operations console, you actually are seeing information from two places:

- ▶ SQL Reporting Services provides the list of reports.
- ▶ The management pack provides metadata such as the report display name and knowledge about the report.

The reports displayed in the Reporting node of the Operations console are actually those found in the SRS root folder. This means that any report is visible, whether or not it is part of a management pack—and is one reason why you want a dedicated SRS for Operations

Manager Reporting (a second reason is the Reporting Component installation changes security such that any existing reports are no longer accessible).

This model also means that you can create a directory in SRS, store a report under it, and have it appear in the OpsMgr console. That makes it easy to add reports, but it doesn't help if you want to deploy a new report to a new management group, deal with localization or dependencies, or easily back up any new reports you have created for Operations Manager. Out-of-the-box, reports are included with management packs, and OpsMgr expects that and deploys them to SQL Reporting Services for you.

There are two steps to backing up custom reports you develop. You can export the report as an RDL file, save it in XML format, and then optionally package it in a management pack. To keep things simple from an administrative standpoint, we recommend that you package your reports in management packs.

Backing Up OpsMgr Reports

To export the custom report as a RDL file, open it in Report Builder and choose File -> Save As File.... This creates the RDL file, which is in XML format. To package the RDL in a management pack, edit it in a text browser and paste its content in an unsealed MP under the Reporting/Report/Definition node in the management pack. As an example:

```
<Reporting>
  <Reports>
    <Report ID="Sample.Report1" Accessibility="Public" Visible="true">
      <Definition>
        [RDL XML file content here...]
      </Definition>
    </Report>
  </Reports>
</Reporting>
```

Where:

- ▶ ID is the name of your report in SRS. It must be a unique name within the management pack.
- ▶ Accessibility="Public" allows other management packs to reference this report. If the report is Public, the ID must be unique in your OpsMgr reporting environment.
- ▶ Visible="true" makes your report visible within the catalog.

Many other parameters can be specified within the node. See Chapter 23, "Developing Management Packs and Reports," for more information.

For purposes of backup, we will assume you have saved your RDL files and included them in unsealed management pack .XML files. These should be placed in a folder with your other customized and unsealed management packs to be backed up.

Backing Up ACS Reports

Microsoft does not support including ACS custom reports in management packs, as by default an OpsMgr administrator does not have access to ACS Reporting. Customized ACS reports should be extracted using the Report Builder and stored in a separate directory. By default, the Report Builder saves to your \My Documents folder; we suggest establishing a common location to save all customized ACS RDL files. You would later use the Report Builder to import RDL files back into the SRS environment.

The ReportingConfig.exe utility, found in the installation media under the \SupportTools folder, is used to reload Microsoft-supplied ACS reports. Chapter 15 provides additional information.

Backing Up SQL Reporting Services Encryption Keys

The SQL Server Reporting Services setup process creates encryption keys that are used to secure credentials, connection information, and accounts used with server operations. If you should need to rebuild or repair your SRS installation, you must apply the key to make the ReportServer database operational. If the key cannot be restored, database recovery will require deleting the encrypted data and respecifying any values that require encryption.

The RSKeyMgmt.exe utility can be used to extract a copy of the encryption key from the ReportServer database. The utility writes the key to a file you specify and scrambles that key using a password you provide. This file should be backed up as part of your backup and recovery procedures. You should also document the password used for the file. Table 12.6 lists the parameters used by RSKeyMgmt.

To create a backup of the encryption key, use the following syntax:

```
RSKeyMgmt -e -fC:\Backups\rsdbkey.txt -p<password>
```

You would run this locally on the computer hosting the report server. The SQL Server Reporting Services Books Online discusses managing the encryption keys under “Managing Encryption Keys.” Michael Pearson has written an excellent article discussing SRS Recovery Planning, available online from the SQL Server Central community (SQLServerCentral.com) at <http://www.sqlservercentral.com/columnists/mpearson/recoveryplanningforsqlreportingservices.asp>. This article discusses SQL Reporting Services 2000, but it is also applicable to the 2005 version.

TABLE 12.6 RSKeyMgmt Parameters

Parameter	Value	Description
-e		Extract a key from a report server instance.
-a		Apply a key to a report server instance.
-d		Delete all encrypted content from a report server database.
-r	installation ID	Remove the key for the specified installation ID.
-f	file	Full path and filename to read/write key.
-i	instance	Server instance to which operation is applied; default is MSSQLSERVER.
-j	join	Join a remote instance of report server to the scale-out deployment of the local instance.
-l	list	Lists the report servers announced in the ReportServer database.
-p	password	Password used to encrypt or decrypt key.
-s	reencrypt	Generates a new key and reencrypts all encrypted content.
-t	trace	Include trace information in error message.
-u	user name	User name of an administrator on the machine to join to the scale-out deployment. If not supplied, current user is used.
-w	password	Password of an administrator on the machine to join to the scale-out deployment.

Automating the Backup Process

As an aid to backing up the files discussed in this chapter, we have created a simple batch file that automates the process from the command line. The backup scripts in the batch file are intended as an example and can be customized for your own environment.

The batch file (backup.bat) is on the CD included with this book. Running the batch file has the following requirements:

Prerequisites:

- ▶ Functional Operations Manager 2007 environment
- ▶ PowerShell installed on the RMS

Installation Steps:

1. Copy the full “\backups” folder on the CD accompanying this book to each OpsMgr 2007 server (including database servers). (This content should include backup.bat, exportmp.ps1, and savekey.exe.)
 2. On the RMS, copy SecureStorageBackup.exe to the %ProgramFiles%\System Center Operations Manager folder from the installation media within the \SupportTools folder.
 3. Customize the script to enable the installed components on the server for each OpsMgr 2007 server.
 4. Schedule the backup.bat program to run according to your schedule (daily is recommended).
-

Disaster Recovery Planning

Although we hope you never need to restore Operations Manager from a catastrophic failure, you must be prepared for the possibility that this could happen. You should have a well-documented recovery plan that would work for every conceivable type of disaster that could occur, from hardware failures to a total datacenter loss. Essentially, you want to be able to get OpsMgr up and running with minimal data loss.

Your plan should assume the worst but be able to concisely and efficiently restore Operations Manager at a minimum to the last backup of your databases. You need to not only develop a detailed plan for the various contingencies, but should also practice the various scenarios in a development environment until you (and others on your staff for when you are not available) are comfortable with the process.

There are at two potential scenarios for disaster recovery, discussed in the next sections.

Recovering from a Total Loss

What would it take to recover OpsMgr assuming a “total loss?” Assume the following scenario:

- ▶ The Operational database is installed on the RMS.
- ▶ The management server is monitoring 200 agent-managed systems.
- ▶ There is only one management server in our management group.
- ▶ The Web console is installed.
- ▶ OpsMgr Reporting and ACS are not installed.

Although this is a very simple implementation of Operations Manager, it is intended to show you the steps necessary to recover OpsMgr from a complete hardware failure of the management server. We will assume that our server team has already built a new server using the same NetBIOS name in the same domain, installed SQL Server 2005, and enabled IIS because we will use the OpsMgr 2007 Web console. The appropriate level of service packs and security patches are applied—be sure to be at the same level of software maintenance that you had with your original system. We are ready to recover Operations Manager.

At a general level, here are the steps involved:

1. Install Operations Manager 2007 from the installation media—selecting the option for a typical installation and using the same management group name as the original install. Remember that the group name is case sensitive. Specify the same accounts (SDK and Config service, Management Server Action account) as used by your original installation.

This type of information should be documented as part of your disaster recovery planning. Detailed steps on installing OpsMgr can be found in Chapter 6, “Installing Operations Manager 2007.”

2. After Operations Manager is installed, immediately stop the SDK service to prevent the RMS from sending data to the Operational database. This prevents OpsMgr from writing data to this database, which you will be overlaying as part of your recovery process. Because any data written to this new database will be lost, immediately really means immediately!
3. Install any additional hotfixes previously installed with your original installation.
4. Delete the OperationsManager database created from your OpsMgr installation in step 1.
5. Restore the latest OperationsManager database created from your SQL backup.
6. Restore the RMS encryption keys.
7. Import any additional management packs that were loaded to your old management server or changed and backed up after your last Operational database backup.
8. Install the Web console.
9. Start the SDK service. Operations Manager will now be functional.

These steps constitute a high-level process for recovering Operations Manager. Your actual plan should contain greater detail, including specific hard drive configurations, the exact installation options, the SQL steps necessary to delete and restore the databases, and so forth.

Using Log Shipping

Another approach for disaster recovery is to implement log shipping. As we discuss in Chapter 10, log shipping automates the process of backing up database transaction logs and storing them on a standby server. This process keeps your production and standby SQL Server systems in synch. Figure 12.23 illustrates a sample disaster recovery solution that includes log shipping for the Operational and Data Warehouse databases.

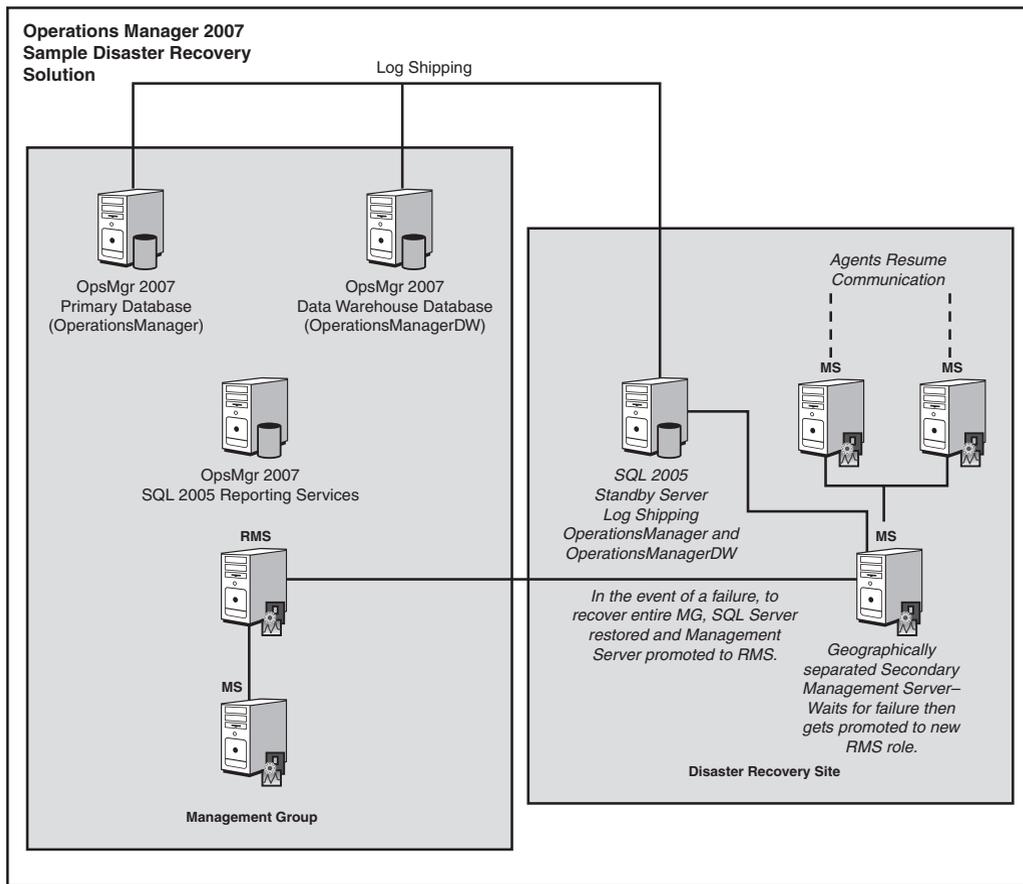


FIGURE 12.23 A sample disaster recovery solution using log shipping.

In addition to deploying log shipping, you will need the RMS and SRS encryption keys for a successful recovery. If you have the OperationsManager database without the RMS key, you will not be able to restore the management group (unless you have SP 1 installed and use the NEWKEY option previously introduced in the "Recovering from a RMS Loss" section of this chapter). The steps to recover from a downed RMS are discussed in the next section.

Recovering from a Downed RMS

Another potential scenario to discuss is if you only lose one of your OpsMgr servers. In this example, we will consider the steps to take if you lose the most important component, the RMS. If your RMS is not available, OpsMgr is not functional. If your RMS is down and you will not be available to meet your SLAs, you will want to promote an existing management server to become the RMS, as depicted in Figure 12.24.

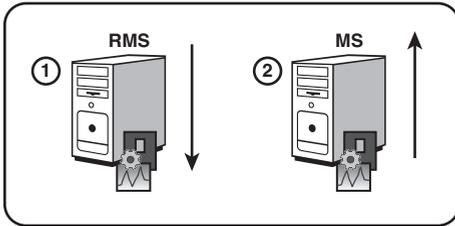


FIGURE 12.24 When the RMS goes down, you can promote an existing management server.

Recovering a downed RMS requires that you have previously backed up the RMS encryption keys, as we discuss in the “Backing Up the RMS Encryption Keys” section of this chapter. You would then promote a functional management server to become the RMS, using the steps we discuss in the “Recovering from a RMS Loss” section. Note that you cannot move from a non-clustered RMS to a clustered RMS, or vice-versa.

A Virtualization Plan for Disaster Recovery

An additional approach for disaster recovery planning is virtualizing your disaster recovery (D/R) environment. This concept would take backups of the physical drives you used when installing and configuring Operations Manager, and convert them to virtual drives.

The advantage of virtual drives is they are hardware independent of the physical environment they run on, making them easy to bring up in a D/R site. You could create a D/R management server in place, maintain a copy of the RMS encryption key, and establish an empty SQL Server(s). If you need to recover your systems, you would restore database backups, promote the management server to become the RMS, and connect the systems. This scenario would work in any software environment supporting virtualization.

Another approach to use virtualization to provide an off-site disaster recovery solution would be through sending regularly scheduled backups of the virtual hard drives to the disaster recovery location. In the event of a disaster, the backup copies of the virtual hard drives are activated and IP address changes made to reflect their new physical location.

Inventorying Your OpsMgr Configuration

Part of any successful disaster recovery plan includes understanding your current configuration. The Operations Manager 2007 Resource Kit includes a tool to assist in taking an inventory of the components changed on each computer where you install an OpsMgr component. This tool, Operations Manager Inventory, collects information about your installation and saves it to a XML-formatted .cab file. Data collected includes the following:

- ▶ Windows Installer logs for Operations Manager 2007
- ▶ Registry information for Operations Manager 2007
- ▶ Operations Manager 2007 configuration information
- ▶ Management packs
- ▶ All running processes
- ▶ All Windows NT event logs on that system
- ▶ The report produced by the Prerequisite Checker when you installed Operations Manager 2007

The inventory tool (MOMInventory.exe) must be run locally on each computer. The computer must have Microsoft .NET 3.0 installed. To run the inventory tool, perform the following steps:

1. Open a command prompt (Start -> Run -> and type **CMD**) and type **MOMInventory.exe**.
2. A dialog box will appear. You can click either Run Collection or Close. Click Run Collection.
3. In the Save As dialog box, enter a name and location for the .cab file the tool will create.
4. While the tool is running, a status window is open, showing information about the data that is collected. When the tool is finished, the status window provides the name and location of the .cab file (see Figure 12.25).

You can also run the tool in “silent” mode. At the command prompt, type the following:

```
mominventory.exe/silent/cabfile:<drive>:\<folder>\<filename>.cab
```

Figure 12.26 shows sample content of the .cab file created by executing MOMInventory.exe.

You can download the OpsMgr 2007 Resource Kit utilities from the System Center Operations Manager TechCenter at <http://go.microsoft.com/fwlink/?LinkId=94593>.

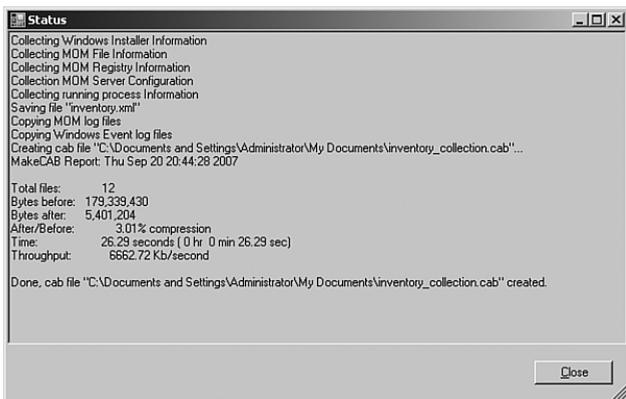


FIGURE 12.25 The MOMInventory tool status window.

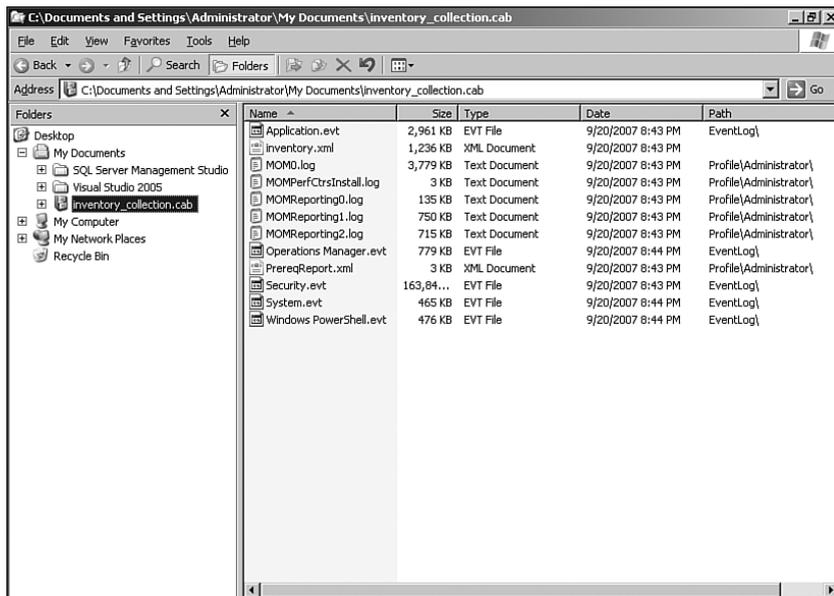


FIGURE 12.26 CAB file contents from running the MOMInventory tool.

Summary

This chapter discussed the need for backups, the components to back up regularly, and the tools available for performing backups. We also discussed an approach for backing up the data warehouse and overall disaster recovery planning for Operations Manager. The next chapter covers administering management packs, including best practices for implementing management packs, and Microsoft resources to help with management pack administration.