



Microsoft Exchange 5.5

Messaging and Collaboration for Demanding Business Needs

Disaster and Recovery Planning

White Paper

Abstract

This paper presents a backup disaster-recovery strategy designed to protect your investment in Microsoft® Exchange Server and to help meet user expectations of 24/7 service and minimal system downtime. To summarize, the paper covers Microsoft Exchange data, where it resides and how it is saved and transaction logging and its application in disaster recovery.

In slightly more depth, this paper covers the built-in backup and restore support of Microsoft Exchange Server and presents a couple of data-recovery scenarios. It explains the steps to take to minimize system downtime if you do encounter a disaster. Some best practices or useful practices are described that can help you administer your servers. The paper also introduces some new features in Exchange 5.5 designed to help with disaster recovery.

© 1997 Microsoft Corporation. All rights reserved.

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

Microsoft, BackOffice, the BackOffice logo, Outlook, and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Other product and company names mentioned herein may be the trademarks of their respective owners.

*Microsoft Corporation • One Microsoft Way • Redmond, WA 98052-6399 • USA
1197*

CONTENTS

INTRODUCTION.....	1
THE BASICS OF BACKUP AND RECOVERY	3
MANUAL BACKUP AND RESTORE.....	4
Online Backup	4
Offline Backups	5
Issues of Concern	5
When It's Time to Restore	6
AUTOMATING BACKUPS USING NTBACKUP.EXE	7
EXAMPLE SCENARIOS	8
Best Practices	9
Minimizing Downtime	9
DISASTER PLANNING AND RECOVERY SUPPORT IN EXCHANGE 5.5.....	11
CONCLUSION	12
For More Information	12

INTRODUCTION

To understand the concept and practice of planning for data disaster and recovery under Microsoft® Exchange, consider the way Microsoft classifies such data. In Exchange, data is classified as either server-based or local. In server-based data, there is the directory service, the information store, the key-management server data, the Microsoft Windows NT® operating system registry, and other files and directories used by Microsoft Exchange.

In the directory service, the data is saved in a file called DIR.EDB, which by default is located in the directory called DSADATA under the default server directory. In addition to the DIR.EDB file, the directory service contains transaction logs, which can reside in the DSADATA directory, in another directory, or on another drive, depending on whether you are running the performance optimizer.

The information store consists of the private information store and the public information store. The private store, which resides in the file PRIV.EDB that's saved by default in the MDBDATA directory, contains user information such as messages and folders. The public information store, which resides in the file PUB.EDB, also by default in the MDBDATA directory, contains public folders.

Like the directory service, the information store consists of the database files and the transaction logs. These logs also reside by default in the MDBDATA directory along with the database files, but if you run the performance optimizer, these transaction logs could be on another drive altogether.

The key-management server is a component that many people keep in mind when formulating their disaster-recovery or backup strategies. Note that in Exchange 4.0 and Microsoft Exchange 5.0, you must install the key-management server as a separate component. This is because the server is not built into the server setup. So, when you install the key-management server, it creates a directory called Security in Exchange 4.0 and Exchange 5.0.

Exchange 5.5 treats the key-management server differently. In that version, the information contained in it is saved in a KMSDATA directory under Exchange. So you should keep the key-management server information in mind when you're formulating your backup strategies.

The Windows NT Registry is also crucial to Exchange Server. Note that the registry contains service-related information as well as most of the configuration information for certain connectors. For example, the Microsoft Mail connector uses the registry to store different PCMTAs you may have set up.

Other directories that may need backup reside under the EXCHSRVR directory, the MTADATA directory, which contains messages and transits through the MTA; and the IMCDATA directory that's used by the Internet Mail Service. In addition to serving as a temporary storage area, the IMCDATA directory stores protocol logging when message archival is on. The TRACKING.LOG directory contains message-tracking files, and the DXADATA directory contains information regarding directory synchronization between Microsoft Mail and Microsoft Exchange.

Local data is ordinarily assumed to reside on the client machine, and the client is responsible for backing it up. Such data typically resides in four areas: personal message stores, offline message stores, personal address-book files, and Microsoft Schedule+ data files. For example, users of Microsoft Outlook™ desktop information manager may store their calendars on their local drive or on their server, depending on their default delivery location.

THE BASICS OF BACKUP AND RECOVERY

This section defines transaction logs and checkpoint files and their respective roles in backup and recovery, circular logging, and transaction logs in playback. First, however, consider the typical message flow within a Microsoft Exchange environment. The Exchange client sends a message to the server, the server receives it, executes transactions that must take place in memory, and almost instantly writes those transactions to a log file. After a certain interval, the transaction is written out through the information store or the database file, the PRIV.EDB or the PUB.EDB.

In other words, for performance and reliability reasons, the transactions are written to the sequential log files first and then to the database files. This means that for every transaction written out to your database file, there is a copy in a log file, which can be played back into your database file in case of a crash. That's the key benefit of transaction logging.

One useful characteristic of an Exchange Server transaction log is its size: whether full or empty, it almost always is 5 MB. So, if you see a log file of some other size, you can assume it is corrupt. The current transaction log is always called EDB.LOG. When it is filled, it is renamed to EDB0001.LOG, EDB0002.LOG ..., and a new EDB.LOG is created. In addition, each transaction-log file contains a signature that must match the signature of the corresponding database file. If these signatures do not match, the corresponding service fails on startup and the event log contains a Jet-level error message, indicating an invalid log signature or an invalid database signature.

The checkpoint file is an optimization, enabling the service to track which transactions have and have not been committed to the database. This file is called EDB.CHK, and for the directory service, it resides in the DSADATA directory. For the information store service, it resides in the MDBDATA directory. Again, if you run performance optimizer, the location of this file may vary. Every time you commit a transaction to the database file, the checkpoint file is updated.

Circular logging is a very important concept for disaster recovery. When circular logging is turned on, it saves storage by preventing the continuous buildup of transaction-log files on your drive. The downside, of course, is that with circular logging, incremental and differential backups do not occur and, therefore, are not available in case of a crash. Note that circular logging is the default setting in Exchange; if you do not want it, you can turn it off through the admin program.

Here's how these transaction-log files and checkpoint files work for recovery. With all your transactions in a transaction-log file and the checkpoint file indicating what transactions have been committed to the database, the service scans the checkpoint files to find the last transaction committed to the database. The service then scans the transaction logs to find the transactions that are not yet committed to the database and writes them to it. This process occurs automatically when you start the service or when you've restored an online backup.

MANUAL BACKUP AND RESTORE

Microsoft Exchange classifies backups as either online or offline. An online backup is made while the Exchange services are running. To back up the data while the services are running, you need an Exchange-aware backup program, such as NTBACKUP.EXE, which ships with Exchange, or a third-party solution. Such programs back up data logically, that is, all the data related to the information store and all the data related to the directory service. You need not tell the backup software to back up, for example, your DIR.EDB and your transaction logs, because the backup program does this for you automatically.

Online Backup

Exchange supports four kinds of online backups: normal or full, copy, incremental, and differential. A normal backup backs up your database files and then the transaction-log files; then it deletes the transaction-log files from the directory. This means you can have circular logging disabled, because your backup software deletes the log files. So if you're performing regular backups, you won't have a problem with log files filling up your drive. To restore a normal backup, you need only restore your last normal backup set and start the service.

A copy backup is similar to a normal backup except that it doesn't purge the log files on your drive and doesn't update the backup context in the database files. Thus, it is practical when you don't want to disturb your normal backup schedule, but you do want to back up your data.

An incremental backup works only on the log files and, thus, only when circular logging is disabled. Like a normal backup, incremental backup also purges log files after backing them up. So it provides yet another way to rid log files from your drive without compromising recoverability. To restore an incremental backup, you must return to your last normal backup set, which contains your database files. Restore those database files, restore every incremental backup set made after the normal backup, and then start the service. Note that you should not start the service until you have restored all the backup sets; otherwise, any logs restored after the backup set will not be played forward.

Like an incremental backup, a differential backup also works on log files, so to use it, you must have circular logging disabled. Unlike an incremental backup, however, a differential backup does not delete the log files. To restore a differential backup set, return to the last normal backup and restore your differential backup set, which contains the entire log files generated after your last normal backup. As with incremental backup, do not start the service until you have restored all the backup sets.

As you might expect, the time required to perform online backup can differ dramatically based on the kind of backup and how often it is performed. If you perform a normal backup daily, the required time is the maximum and remains constant each day. If you perform a normal backup on Friday and an incremental backup Monday through Thursday, the required time also is constant, but is much lower than that required from a normal backup each day. If you perform a normal backup on Friday and a differential backup on each of the remaining days, the

required time is lower than that required for a normal backup, but increases from Monday through Thursday, because you are not deleting the log files.

The advantage of a daily normal backup is simplicity: It's the easiest backup to schedule and to restore, because you need only return to your last normal backup. The disadvantage is that it requires the maximum time each day and, depending on the size of your database files, may require that you swap tapes each day.

The advantage of a normal-with-incremental backup is that it requires the minimum backup time and thus exerts the least impact on the server. The disadvantages are that you must perform between two and seven restores in the case of a crash, a situation requiring potentially multiple tapes, and you must have circular logging disabled.

The advantages of a normal-with-differential backup are that it exerts less impact on the server than a daily normal backup and, more important, it provides multiple copies of logs for most of the week. For example, if you find on Saturday that a log file generated on Monday was corrupt, you can return to any one of the week's previous daily backups and restore that log file. The disadvantages are that you need two restores in the case of a crash, your backup time increases every day, and you must have circular logging disabled.

Offline Backups

An offline backup is a normal file-level backup made with services stopped. Any backup software can perform an offline backup. However, when you restore, an offline backup does not automatically play through the log files as does its online counterpart. For this reason, Microsoft does not recommend an offline backup for daily backups. Nonetheless, an offline backup is essential when online backups fail.

Issues of Concern

Whatever kind of backup you're performing, keep in mind the respective roles of the key-management server and the Windows NT Registry. When you back up the directory service and information store, the key-management server data is not backed up. To back up this data, you must stop the service and back up SECURITY\MGRENT (under either Microsoft Exchange 4.0 or 5.0) and KMSDATA5.5 (under Exchange 5.5). If you select the option to use the key-management server startup disk, you must back it up regularly as well. Without a copy of this disk, the service will not start.

As far as Exchange is concerned, services and connector configuration reside in the Windows NT accounts database, which itself resides in the Windows NT Registry. If someone should accidentally delete your service account or if you should find the database to be corrupt, you may need to restore it. Now, you cannot recreate the Exchange service account, because the SIDS are different between the original account and the new one; and if you don't have a backup of your accounts database, your only option is to reinstall every server that uses that service account.

So it's essential that you do this backup, especially on your domain controllers.

When It's Time to Restore

Of course, the way you restore data depends on whether you're returning to an online or an offline backup, and restoring an online backup is easier than restoring an offline backup. In a daily normal backup, you simply restore the last normal backup and start the service. In a normal-with-incremental backup, you restore the last normal backup and all incremental sets and start the service, and Exchange plays through the log files. In a normal-plus-differential backup, you restore the last normal backup, restore the last differential backup, and start the service.

In an offline backup, you follow one procedure for restoring the directory service and another for restoring the information store. For the directory service, restore the DSADATA directories, using the Windows NT Registry if necessary to locate multiple DSADATA directories on different drives. Then, start the service. For the information store, restore the MDBDATA directories (whose locations also are shown in the Registry) and then run a program called ISINTEG.EXE found in the bin directory under Microsoft Exchange Server and provide this program the "-patch" command-line option. Then, stop the service, and it should start up again on its own.

To restore the Windows NT Registry, run NT.EXE and check the box that says "Restore Local Registry." To restore the key-management server service, you must stop the service, restore the corresponding directory, restore the startup disk, and start the service.

AUTOMATING BACKUPS USING NTBACKUP.EXE

To automate backups, use NTBACKUP.EXE to create a batch file that configures a backup process to do whatever you want. You can back up just the directory service or just the information store using normal, incremental, and differential backups. You can schedule the running of this batch file using the Add Schedule program that's built into Windows NT, the WinAT program that ships with the Windows NT Resource Kit, or a third-party backup solution.

EXAMPLE SCENARIOS

Exchange supports two major data-recovery scenarios: single item and full server. As the names imply, a single-item recovery is called for when someone accidentally deletes a single message, mailbox, public or private folder, and the like; and a full-server recovery is called for when you must recover all the information on your server. To recover a single item such as a message, mailbox, or folder in Microsoft Exchange 4.0 or 5.0, you must have a dedicated server separate from your current Exchange site, capable of running Exchange, and having sufficient disk space to hold a restore of the entire information store.

On this recovery server, install Windows NT and any Windows NT Service Packs that you want, and then install Exchange with the same organization name and the same site name as your production server. Take care in the process that you do *not* join the existing site but, instead, create a new site with the same organization name and the same site name as the original machine. Then, upgrade to the same Exchange Service Pack as the version that was on your production server when you made the backup that you intended to restore.

Now, restore the information store and then run the DSIS consistency checker. Since you haven't yet restored the Exchange directory service, the consistency checker populates the directory service with information it obtained from the restored information store. Then log on to the admin program and assign permissions to the mailbox or public folder you're after, using the Exchange client to log into that mailbox and copy the information to a PST file that you can provide to the user in question.

In Microsoft Exchange 5.5, you can automate much of these steps using a feature called "Item Recovery," which is discussed in more detail at the end of this White Paper. In addition, there may be quick single-item recovery capabilities provided by third parties.

As for full-server recovery, you typically perform such a recovery when your server has been destroyed or retired, as when you upgrade to a more powerful machine. In contrast to single-item recovery, full-server recovery requires that you restore the information store as well as the directory. For this reason, the recovery server must have the same name as your production server. You also must access the same service account as your production server and, in turn, the same Windows NT accounts database. So you must have a domain controller present in addition to the Exchange Server that's being recovered.

First, use the Windows NT Server Manager to delete and recreate the account of the machine being recovered into the domain. Then, install Windows NT Server using the same machine name as your production server, any Windows NT Service Packs you might want, and Exchange Server, and select the option to create a new site, just as in a single-item recovery. Then, upgrade to the same Service Pack that the original server had and, if the original server had a Microsoft Mail Connector, configure it. If the original server had a key-management server, install that. Then, restore the directory service and the information store, restore the key-management server data, and start the services.

Best Practices

There are a number of steps you can take to prepare yourself against a crash. First, it's crucial that you perform daily online backups, that you disable circular logging, and that you verify backups regularly. You also should back up your registry regularly on your domain controllers and occasionally shut down the services and perform a full file-level backup.

Second, perform regular monitoring of your event and application logs, where you can learn whether your online backups are successful. Third, set limits on the sizes of messages and mailboxes so as to help prevent your information store and, in turn, your backup and restore time, from increasing uncontrollably. Fourth, periodically clean out mailboxes, especially the administrator mailbox associated with certain connectors. Fifth, separate databases and swap files from transaction logs, locating the former on a RAID5 stripe set and the latter on a dedicated physical drive. This can significantly reduce the probability that you would ever lose them all simultaneously, and it can improve performance. Sixth, maintain sufficient free-disk space on your drives so that you can routinely run database-defragging utilities.

Minimizing Downtime

No matter what precautions you take, there's always some risk of a crash, so you also should learn how to minimize downtime. Consider this time in two parts: the time required to restore and the time required to prepare the recovery machine. Reducing the restore time requires that you keep backup tapes at an easily accessible site and have a speedy restore solution in place. The Exchange Server 5.5 software, for example, provides the fastest restoration rate now available using tape drives, so the limiting factor is the hardware. Keep restore times in mind especially as you move to Microsoft Exchange 5.5 with its unlimited store.

Minimizing the time to prepare your recovery machine is best done by maintaining a "hot spare"—an onsite machine that's preconfigured to support your Exchange environment. There are two primary issues associated with a hot spare. First, the Exchange directory service is tied to the NETBIOS name. So you cannot have the directory service installed on another machine, because you cannot have two machines on your network with the same NETBIOS name.

Second, the Exchange directory service depends on the service account. So if your hot spare on another network doesn't have access to your domain accounts, then you cannot have a successful restoration and startup of the directory service on that machine.

When you are using a hot spare for single-item recovery, all you must do is restore the information store. To prepare the hot spare, install Windows NT, your Windows NT Service Packs, and Exchange Server, with the same site and organization name and without joining the existing site. Also, be sure that your Windows NT Service Pack is the latest version running on your production server.

To restore the information store, simply run DSIS and assign permissions.

When using a hot spare for full-service recovery, you must restore both the information store and the directory service. Recall that the directory service depends on the NETBIOS name and that you cannot have two servers with the same name on the network. To prepare the hot spare, install Windows NT and give the machine a different name, but join the same domain. Then, install the appropriate Windows NT Service Pack.

Keep in mind that you should use the same hardware as the original server although it's not absolutely necessary. It is necessary, however, that you have an available domain controller other than the recovery server and the current Exchange Server. This is because when your Exchange Server goes down, the other domain controller should be able to provide the accounts database to your recovery server.

Once your Microsoft Exchange Server goes down, delete and recreate the machine account to the domain and rename your hot spare to be the recovery-server name. Now you can install Exchange on it and perform your restoration. This approach reduces the time required to install Windows NT and the Windows NT Service Packs.

For recovery, the steps are the same as discussed earlier. Create a new site with the same old site name, select the same service account, install the same connectors, and upgrade to the same Service Pack. Then, restore the directory service and the information store, reconfigure connectors if you have them, and install the key-management server before you do any restoration. Then, restore the directory service and the information store, restore the key-management server data, and start the services.

Unfortunately, you may have a crash at a time when you don't have a recent backup. In this case, your data loss depends on the kind of backup you do have. If you have a month-old online backup, for example, with circular logging disabled, then you need only restore the last backup and you'll recover all your data. Alternatively, if you have only an offline backup or an online backup with circular logging enabled, then you are likely to have some data loss and will need a database repair—the last resort in disaster recovery.

DISASTER PLANNING AND RECOVERY SUPPORT IN EXCHANGE 5.5

Microsoft Exchange 5.5 provides two primary features designed to expedite disaster recovery: Item Recovery, for single-item recovery, and Microsoft Exchange Cluster Server support, for full-server recovery. Item Recovery is like a recycle bin for the information store, providing you with the ability to recover deleted items. In the private information store, Item Recovery helps you recover deleted messages and folders, although not deleted mailboxes. Similarly, in the public information store, Item Recovery helps you recover deleted messages and deleted folders.

Item Recovery relies on some of the basic design considerations of Exchange 5.5. Every object has a new attribute associated with it. When you delete an item (or folder), that item remains in the information store even as it becomes hidden from your view. Configuration of this is done through the administrative program, and recovery is controlled from the client. To configure this on the server object, you set properties on the private information store determining how long and under what conditions to hold a deleted item. You can set similar properties on an individual mailbox as well, using default or custom settings for an individual mailbox.

As for recovery, Exchange 5.5 includes a new client extension for Item Recovery known as Restore Deleted Items. When you select it from the Tools menu, it displays an undelete dialog similar to that of the recycle bin. There, you can restore messages and folders.

To summarize, Microsoft Cluster Service Support means you can have a group of independent systems that appear as a single system. You can manage them as a single system, and they can use a common namespace. Services are “cluster-wide,” and the cluster provides a high tolerance for component failures. As a result, you enjoy a greater ability to protect against data loss and, in the event of a crash, restore all your data quickly and easily.

Other resources to which you can turn for help in disaster recovery are Microsoft TechNet and its trove of Microsoft PSS Knowledge Base articles; the Microsoft BackOffice® Resource Kit, which also includes recovery-specific utilities; and the Exchange ListServ.

CONCLUSION

Microsoft designed Exchange with recovery in mind, and so, with it, you should be able to recover from virtually any disaster that you might run into. This is especially true if you follow best practices such as performing regular online backups, disabling circular logging, and verifying backups periodically.

For More Information

For the latest information on Microsoft Exchange, check out our World Wide Web site at <http://www.microsoft.com/exchange>.