©1999 Robichaux & Associates. All rights reserved. Visit http://www.robichaux.net

- Understanding Server Recovery
- Backing Up Exchange
- Maintenance and Repair Tools
- Recovering Your Data
- Preventative Medicine

17

Recovery and Repair

I do not believe in recovery.

Lillian Hellman

Scoundrel Time

In some organizations, email has become more important than the telephone-- consider how a business like Microsoft or Amazon.com would operate internally without email. Microsoft and its partners have been pushing hard to make Exchange into a single messaging source, unifying voice mail, fax, and email into a single inbox. This unification makes disaster recovery even more critical.

Disaster recovery is a complicated topic. I'm not going to address the basics of recovering a crashed NT server, nor will I discuss the low-level preparations (like keeping a copy of your backup tapes offsite) you should already be doing. The Exchange documentation makes very little mention of disaster recovery; instead, the BORK and two white papers are usually cited as canon. I'll outline what you need to know about the fundamentals of disaster recovery and explain where you can find specifics for unusual tasks like recovering one system in a cluster.

Understanding Server Recovery

Let me begin with a simple but often-overlooked true fact, which I've formatted as a warning to catch your eye if you're just skimming.

©1999 Robichaux & Associates. All rights reserved. Visit http://www.robichaux.net

If you don't have good Exchange backups your ability to recover your servers will be *extremely* limited. You *must* make sure your backup procedures work, and you must continually monitor the process to ensure that your backups are usable.

Exchange Server's normal backup procedures are primarily geared toward single-server disaster recovery; they're designed to help you replace an entirely failed member server, no matter what caused the original loss. The IS and directory databases are highly specialized, which is what gives them so much functionality, but the recovery and repair procedures necessary to fix a downed server are very specific to Exchange. You can't just pop in a tape, run *ntbackup*, and go.

Exchange is easy enough to set up and configure that it's possible to overlook some fundamental requirements you have to meet to make sure your configuration will be recoverable if something happens to it. There are three key requirements:

- You have to plan your recovery strategy to make sure you have everything you'll need to do a recovery: hardware, software, and backups.
- You have to understand the specifics of your configuration and how to best recover it.
- You have to practice recovery procedures so that when it's time to do a *real* recovery you're not flustered or frazzled by the unfamiliar procedures.

Planning Your Recovery Strategy

Failure to plan is the biggest cause of permanent Exchange data loss. Why? Consider a typical single-site environment with two servers. One server is lost in a fire. Its backup tapes were sitting next to it on a desktop instead of in an offsite (or even fireproof) vault. Oops. That server's not recoverable because its administrator didn't plan adequately, not because the server got roasted. The key points you have to consider when planning your recovery strategy are straightforward:

- How long can I afford for my server to be down? Later in this chapter you'll learn how to estimate backup and restore times; on top of that, you have to factor in the time it takes to run Exchange's database repair tools if needed, plus any time to get hardware set up and organized.
- Do I have adequate replacement hardware? If your primary server is a large quad-processor box with a 60GB store, what happens when you need to restore all 60GB of data to another machine? The best possible outcome is to keep a clone of your standard server configuration in your test lab so that you can use it as a recovery server (that implies that you have to budget accordingly, as discussed in Chapter 3).

©1999 Robichaux & Associates. All rights reserved. Visit http://www.robichaux.net

This also extends to backup hardware. For example, if you back up your servers to DLT, you'd better make sure you have more than one DLT drive, because if your sole drive fails you won't be able to back up or restore until it's fixed or replaced.

- Am I making regular backups? Do they happen often enough to capture all the changes made to my server? Do they include system information like the domain SAM and server registry?
- Are my backup tapes secure? Ideally, you should have multiple sets of backup tapes, some of which should be stored in a secure offsite location. (See the Nutshell Handbook *Windows NT Backup and Restore* by Jody Leber for more details.)
- Am I making the right kind of backups? There's a difference between offline and online backups, and in general online backups are more useful and easier to restore. Do you know what kind you're making right now?

Steal this white paper

Microsoft has prepared a terrific 84-page white paper titled "Exchange Disaster Recovery". It was originally written for Exchange 4.0 by a group from Microsoft Consulting Services; since then, it's been updated several times and is still the canonical source of disaster recovery knowledge. One reason it's so long is that it consolidates in one place much of the information spread throughout this book.

Stop reading this book right now and go get the white paper; it's available from *www.microsoft.com/exchange/55/whpprs/BackupRestore.htm*. Once you've got a copy, come back and resume reading; I'll be making reference to the white paper throughout the rest of this chapter, and it's a very handy reference to keep right next to each of your servers.

Estimating recovery space requirements

Some recovery procedures use more space than others. In particular, any time you have to run the *isinteg* or *eseutil* tools, it's likely that you'll need a significant amount of extra disk space. A good heuristic is that you should always have at least as much free space as your largest store file; some repairs require *twice* as much space as your largest database file, so be prepared to juggle things around (or use a network shared drive, as described later) to leave yourself enough room.

Don't forget that this extra storage is required for most database maintenance tasks, too; if you plan to do offline defrags or inspections with *isinteg*, you'll need the extra space.

Estimating recovery time

To know how long it'll take to recover your system, you first must know how long it will take to physically move your backed-up data to the recovery system. Once your store

©1999 Robichaux & Associates. All rights reserved. Visit http://www.robichaux.net

gets bigger than about 5GB, recovery time is driven primarily by the speed of your I/O subsystem. Even with a Fibre Channel RAID array, an 18GB file copy between two separate controllers and disk arrays can take more than an hour. Over a 100Mbps switched full-duplex network, a 16GB file copy time still takes around 90 minutes. When you consider the sad fact that restoring data takes at least as long as backing it up, and that running the Exchange utilities can push the total recovery time to twice as long as the original backup requirement, knowing how long things take is critical.

It's prudent to establish a baseline for the speed of your network and servers. I recommend the following tests, using at least a 2GB store file:

- If you intend to run *eseutil* or *isinteg* on the same disk partition as the databases, do a disk to disk copy on that partition.
- If your system has multiple controllers, do a disk to disk copy between the two controllers on one server.
- Do a disk-to- disk copy between two unique servers; this will tell you what sort of network throughput you can reasonably expect during a restore.
- Back up your stores from disk to tape using your backup software. For best performance, put your tape drive on a different controller than your database disks. With some systems and backup software, local backups will be faster than doing backups over the network; on others, CPU load and other overhead will actually make this slower.
- Back up your stores from their home on one server to a tape drive on another. This offers you superior flexibility when it's time to restore, since you aren't limited to restoring onto a machine with a tape drive.

These tests take time to run, but one late night or Saturday spent doing them provides real-world data on how long restores will take in your environment. These tests will tell you how long it takes to restore the data, but not how long it takes to play back the log files. There's no really good way to estimate that, since many short transactions take longer to play back than an equal-size block of a few large transactions.

Note that these tests may show that your I/O system isn't well-optimized for copying large files. That's OK, since most Exchange I/O is made up of small (four to 64 KB) requests. See the next chapter for more details.

One interesting fact to note: Exchange 5.5 has been tuned to make the database backup APIs significantly faster. Microsoft claims that Exchange supports speeds up to 30GB/hour, so if you can't back things up that fast the problem is likely that your backup hardware can't keep up.

©1999 Robichaux & Associates. All rights reserved. Visit http://www.robichaux.net

Logging and The Databases

Chapter 2 covers database transaction logging in detail, including a general overview of backups; you may find it helpful to flip back there if you need a refresher. The logs often actually grow faster, and consume disk space faster, than the database files themselves. Every transaction generates a log entry. For example, if you send yourself a message containing a 24MB video clip into your inbox, then delete the message, you'll have generated more than 24MB of log files—one set of log data records the new message, while a separate set records its deletion.

The biggest stumbling block to successful Exchange recovery is circular logging. That's because when you enable it you're giving Exchange permission to throw away log data. With circular logging off, as long as you have a good copy of your transaction logs, you can restore your database to a consistent and correct state.

If being able to recover your data is more important than the cost of having to buy enough disk space, *turn circular logging off on your IS servers*. There's no excuse for leaving it on when you can buy a 10GB drive for US$150 or less.

A recovery scenario

Even if you lose the entire public or private IS (say, the single disk it's on crashes beyond repair), you may still be able to restore the server with *no* data loss. For this to happen, though, there are two ironclad requirements:

- You must have a complete backup-- either a full backup or a full backup combined with appropriate incremental and differential backups. If you use differential backups, *every* one of the differential tapes must be available and complete.
- Circular logging must be turned off, and you must have access to the log files either on their original disk or from a recent backup. (Remember, according to what you learned in Chapter 3, your logs should always be on a separate disk from your IS databases.) Exchange 5.0 and 5.5 turn circular logging on by default, so you must manually turn it off.

As long as you meet these requirements, data loss can be zero. Let's say that the primary server for your workgroup is highly active; it has a 24GB *priv.edb* and logs grow at a rate of about 40MB per hour. Backups are performed once each night: a full backup, starting at 11:00pm and ending around 12:30pm. The logs and databases are stored on separate drives.

At 3:05pm on Tuesday, a cleaning lady is working in the server room and accidentally pulls the power cord out from the back of a server. This happens right when Exchange is

©1999 Robichaux & Associates. All rights reserved. Visit http://www.robichaux.net

updating a table in the private IS, so the IS ends up corrupted. Upon reboot, the IS service fails to start. You review the event log and determine that the IS won't start because the database is corrupted. You opt to restore from your backups.

Although the IS is down, the DS and SA services are running. You start *ntbackup*, pop in last night's backup tape, and select to restore the *priv.edb* from last night and start the restoration. Approximately 70 minutes later the file is restored.

Now, let's stop to consider what's happened so far. Your Exchange server now has a *priv.edb* from 12:30am on Monday, but the server last saw activity at 3:05pm. There are hundreds (if not thousands) of new messages, sent items, and user changes to the database that happened in those 14 hours and 35 minutes. There's good news, though: since you still have pristine copies of the log files, once you restart the IS it begins to play back the approximately 560MB of log data that accumulated since the backup finished. Each transaction in the log is re-recorded in the private IS, so when the IS finishes its job your store is returned to its exact pre-unplugging condition.

How likely is this scenario? Very! If the drive with your IS had failed, or if you had to repair some other component, you'd still be able to restore things the same way. Even if you have to rebuild a server from scratch, as long as you have good backups you'll be able to make things right. Note that this recovery wouldn't have been possible without proper planning and execution, and it still required time to do the recovery itself.

Depending on what caused the original corruption, it's conceivable that replaying the logs will just lead to the identical circumstances and again corrupt the store. Be sure to pin down the cause of failure before you try to recover from it.

It is worth noting here that a $200 call to PSS for a helpful walkthrough of the recovery process may be the cheapest insurance you ever buy. If you don't practice recoveries regularly, and if you don't have a detailed plan written by you and customized for your environment, it is cheaper to get help beforehand than after a bungled restoration.

Backing Up Exchange

Exchange is designed to be backed up while it's running. These online backups give you maximum access to your data, since you don't have to stop Exchange's services to do a backup—provided you use an Exchange-aware backup product. (See the sidebar "Third-Party Backup Tools" for more on that touchy subject.) Before you start making backups, though, you need to know what to back up.

©1999 Robichaux & Associates. All rights reserved. Visit http://www.robichaux.net

What To Back Up

How do you know what to back up? You can go the easy route and just back up *everything* on your server; even then you still need to understand what specific items are most important. You may also want or need to be more selective about what you preserve on your backups. Here's a suggested list:

- The public and private IS databases and the directory database. To do a successful online backup you'll have to use an Exchange-aware backup tool.
- Transaction logs. The public and private databases share one set of logs, so they are backed up together, but the directory logs are separate.
- The Key Management Server database. You must stop the KMS before you back up its data, then restart it when you're done. (Remember that the KMS data is stored in its own directory, and that it's not backed up as part of the Exchange backup process.)
- Other miscellaneous files in the Exchange directory trees, including MS Mail and other connector files, the GWART files, IMS archive messages, and so on.
- The Windows NT registry on the Exchange server. *ntbackup* includes a checkbox you can use to specify that you want this backed up; you should also keep your emergency repair disks current.
- The Windows NT SAM database. Exchange depends on Windows NT security information for the service accounts and user access to mailboxes, so to restore a server you must have access to the SAM context it formerly lived in.
- If you're allowing your users to use PST or OST files, and if you're storing them on a central server, be sure to back them up too. Better yet, don't allow your users to use them.

Backup Considerations

Exchange Server supports three forms of backup: full (also known as normal), incremental and differential. If you insist on using circular logging, you can only do full backups; incremental and differential backups depend on the log files.

Full

A full backup, as the name implies, is a complete backup of the data; everything you specify is backed up in its entirety. All Exchange recovery depends on having a full backup plus any incremental or differential backups you use. Since a full backup captures everything on your server, it has the advantage and disadvantage of being comprehensive—a full backup is complete, but each backup requires the same amount of time.

©1999 Robichaux & Associates. All rights reserved. Visit http://www.robichaux.net

Exchange-aware backup applications purge the transaction logs after the backup completes; that's safe because by the time the backup finishes all of the logged transactions have already been applied to the store and backed up to tape.

Differential

Differential backups capture only those changes since the previous full backup. Exchange-aware products implement differential backups by storing only the log file that have changed since the previous full backup, not any of the IS or DS database files themselves. Differential backups don't purge the log files.

Because differential backups preserve all changes since the previous full backup, you don't have to keep all of them together. Let's say you do a full backup every Sunday and differentials Monday through Friday. You can use the full backup plus any one of your differential backups to do a complete restoration. This means that as the week progresses your differential backups will take longer.

Incremental

Incremental backups are similar to differential backups, with a major twist: at the end of the backup, the transaction log files are purged. This is a lot more aggressive than differential backups, since if anything goes wrong with your backup media you no longer have log files! Incremental backups have to be used together; if you make a full backup on Sunday and incrementals the rest of the time, restoring a server that fails on Wednesday requires the full backup plus Monday and Tuesday's incrementals.

Choosing the right backup strategies

Instead of focusing on saving tapes or labor in doing the backup, focus on saving yourself trouble and effort when you need to restore your servers. That's when the availability of the right backup data really pays off. If doing a daily full backup takes too long or uses too many tapes to be feasible for your site, get new backup hardware. When the time comes to do a restore, the last thing you want to have to worry about is chasing down all of the incremental or differential tapes you need.

If you can't do full backups every day, do a couple per week and do differentials on the other days. If your disk space situation forces you to use incremental backups, you can do so, but I'd only recommend it until you can get a bigger drive for your log files.

Safeguarding your backup tapes

While a complete discussion of the pros and cons of tape rotation and offsite storage doesn't belong in this book, I can give you an abbreviated version: use three sets of tapes, which I'll call A, B, and C. Each week, rotate the sets. For example, if you use A this week, then B should be in a fireproof vault at your facility and C should be in a similar

©1999 Robichaux & Associates. All rights reserved. Visit http://www.robichaux.net

vault in a different location. Next week, you'd start using C, you'd move B offsite, and you'd put A in your office vault. This shuffling may seem like a terrific hassle, and it is, but it's also excellent protection against losing or damaging one set of tapes.

I also strongly recommend that you establish a periodic audit of your backup procedures and their implementation. Check to make sure that your scheduled backups are actually happening, that there's data being written to the tapes, and that you can restore it again. I once was called in to restore a system that had failed; the system's tape drive was slightly out of alignment, so that it could read tapes it had written, but other drives couldn't. That put a quick stop to the restoration until we figured out what was wrong.

Content considerations

In some environments, email requires special attention for backup procedures and data recovery ability. For legal and business liability reasons, efforts may be made to ensure that long-term backups are not retained. In the United States of America, it is not uncommon to hear of companies who have policies on how long tapes may be kept. The contents of email "conversations" can be used in legal proceedings; everything from sexual harassment cases to establishing that an organization had a specific intent during dealings with a partner company.

Using *ntbackup*

Exchange includes a modified version of the *ntbackup* utility. It's been modified to understand how to find Exchange servers in an organization, connect to them, and back up their data without stopping the server's Exchange services. When you install Exchange or Exchange Administrator on a machine, you get this modified version of *ntbackup*. You can tell when you have it because there will be a new Microsoft Exchange… command in the Operations menu.

What *ntbackup* can do

It's not as sophisticated as some of the available third-party products, but *ntbackup* is still my favorite solution for backing up Exchange servers. That's because its integration with Exchange is seamless—to back up a server's DS or IS, all you have to do is pick the server and items to back up, then sit back and relax. You don't have to stop your Exchange services or otherwise fiddle with things.

ntbackup supports a wide variety of IDE and SCSI tape devices, but it doesn't support removable media or hard disks, so you can't use it to do an online backup to another disk somewhere. In addition, it doesn't have a lot of fancy tape management features. It doesn't know about autoloaders, striped DLT arrays, or other exotica. However, it's robust, reasonably fast, and free.

©1999 Robichaux & Associates. All rights reserved. Visit http://www.robichaux.net

Using the GUI

ntbackup is pretty easy to use, so I'm not going to spend time explaining its basic functionality. The online help is good, and you can learn a lot from just poking around. Instead, let's dive into the Exchange-specific changes.

When you run *ntbackup* and select the Operations-->Microsoft Exchange… command, you'll see a window split into two panes, just as in Exchange Administrator. The left pane shows your organization, sites, and servers. When you select a server in that pane, the right pane shows what items you can back up for that server. The DS and IS are shown separately, but no distinction is made between the public and private IS files.

To specify a server or database to back up, just click the checkbox next to it. Once you've selected everything you want to back up (bearing in mind that recovery will be easiest if you back up each server separately), you can click the Backup button or use the Operations-->Backup… command to start the festivities. The familiar Backup Information dialog will appear; it looks just like *ntbackup*'s standard dialog, except that you'll see each server's DS or IS as a separate backup set labeled "Microsoft Exchange" plus the server name (for example, "Microsoft Exchange: Information Store \\HSV1"). You can use the Backup Type pulldown to select a backup type (either normal, incremental, or differential). If you're running *ntbackup* on the Exchange server itself, you can use the Backup Local Registry checkbox to force it to back up your registry as part of the session. When you're done specifying what you want backed up, just press the OK button and the backup will begin. As things progress, the Backup Status window will show you which organization, site, and server it's backing up, and it will log any errors it finds.

> Don't assume that your entire backup will fit on a single tape—it might not, and *ntbackup* will dumbly sit there waiting for the next tape. In the meantime, incoming transactions will pile up in your Exchange server's database.

Exchange-specific command-line switches

ntbackup has a bunch of command-line switches, all of which are well-documented in its online help. Oddly, the Exchange-specific switches aren't documented at all. Since you need them to do command-line backups, this is a pretty major omission, but there are only two pertinent switches: `DS` and `IS`. These work like you'd expect: in your *ntbackup* command line, you use the switches along with the machine name you want to back up. For example, this command line backs up drives *c*, *d*, and *e*, along with the DS and IS on the server named *HSV*:

```
ntbackup backup c: d: e: DS \\hsv IS \\hsv /v /d "HSV DS/IS/drive" /b
     /t Normal /l c:\backup\daily.log
```

©1999 Robichaux & Associates. All rights reserved. Visit http://www.robichaux.net

Testing your backup

You can, and should, verify the integrity of your backups. Microsoft's recommended approach is covered in KB article Q178308; in brief, they recommend that you do the following:

1. Set up a test server that you can restore the backup onto.
2. Restore the backup onto your test server.
3. Stop the IS and DS services on the test server.
4. Use the *eseutil /g* command (described below) to verify the integrity of the restored data. You'll need to run it three times, once for each of the database files.

If the restore was successful, none of the above steps will report any errors. I also recommend some basic sanity checks on the database content—you might consider checking mailboxes or public folders with known content to be sure all that content is there. You could also use Outlook's Advanced Find command to search for random strings on the entire mailbox, searching both subject and body; this is a good test of the IS' random-access indices. When you're done, check the test server's application event logs for any unusual errors.

Restoring from your backups

Restoring from a backup made with *ntbackup* is pretty easy; it's the other steps involved in the recovery process that are tricky, like knowing when to start the Exchange services. To restore from an online backup, run *ntbackup*. When the Tapes window opens, double-click your backup tape (or use the Operations-->Catalog… command) to force it to catalog the tape. Once the catalog's been loaded, use the Tapes window to check the databases you want to restore, then use the Operations-->Restore… command to start the restore. The Restore Information dialog will appear, and you'll need to fill it out properly:

- You can only restore a DS to the server it was backed up from; by filling out the Destination Server field, you can restore an IS to another server.
- You can restore the private and public IS databases separately by checking the appropriate boxes.
- If you check the "Erase all existing data" checkbox, *ntbackup* will replace the existing databases on the target server with the new ones you're restoring. Use this option only when you're sure you want to overwrite what's already on the server.
- The "Start Service After Restore" checkbox lets you choose whether to automatically restart the DS or IS service once the restore finishes. Don't check this unless you're sure you don't need to run *isinteg*.

©1999 Robichaux & Associates. All rights reserved. Visit http://www.robichaux.net

Once you've set the restore options the way you want them, click the OK button and the restore will start. Since *ntbackup* has to stop the Exchange services before doing the restore, it will ask you to confirm that you want them stopped.

Offline Backup

If the Exchange IS service is cleanly stopped, then the *priv.edb* and *pub.edb* files will be closed normally and all logged transactions will be correctly posted. Once you've done this, it's possible to save the *priv.edb* , *pub.edb*and *DIR.EDB* files to disk or tape, in effect making a backup using the filesystem as a backup tool. Since the Exchange services aren't running while you do this backup, it's called an *offline* backup.

Generally online backups are best; they preserve your ability to back up your data while still keeping your server operating. However, there are circumstances where offline backup and recovery are useful, as when you're constructing an alternate server or running tests that require a complete copy of the database files.

The critical ingredient to doing a successful offline backup is to make sure that the IS is correctly shut down. If the stop attempt failed, or if the service stopped as the result of a crash, then the database files are not suitable for an offline backup. At a minimum, you should restart the IS and stop the service correctly before making a copy of the files.

If you're using a third-party backup solution that doesn't support Exchange, you may be tempted to rely on offline backups as your primary safety net. I don't recommend it. Offline backups don't delete the log files as online backups normally do. In addition, Exchange-aware backup programs run a page-level integrity check on the database as each page is backed up. When an offline backup is performed, the database pages are not checked-- the file is just copied, so database damage can go unnoticed (at least until you need to restore from your offline backup!)

> Of course, if you keep daily watch on your event logs you'll notice any unusual developments associated with the backups before they can do permanent harm. Microsoft Knowledge Base article Q188646, titled "XADM: Unable to Back Up Exchange Server 5.5 with Event ID 105", explains what to do when you encounter underlying database problems when you're doing a backup.

Restoring from an offline backup

There are four key steps to successfully restoring from an offline backup:

©1999 Robichaux & Associates. All rights reserved. Visit http://www.robichaux.net

- Safeguarding the existing database files, just in case you want to undo the restore. To accomplish this, copy the contents of the *exchsrvr\mdbdata* directories on all your system drives to a safe location.
- Finding the correct location for the IS and DS databases and logs. These paths are stored in the registry: the `HKLM\System\CurrentControlSet\Services\MSExchangeIS` key has separate entries for the database log path (`ParametersSystem\DB Log Path`), the public IS database (`ParametersPublic\DB Path`), and the private IS database (`ParametersPrivate\ DB Path`); there are separate entries for the directory in the `MSExchangeDS` key.
- Copying the files from wherever you backed them up to back to the correct location. The public and private IS databases, IS log files, directory database, and directory logs can all be in different directories, and it's critical to get them in the correct location.

You can't restore log and database files to a different path than the ones they came from, because they contain internal signatures—you can, however, restore them to different drives, as long as the relative path remains the same.

- Running *isinteg* `-patch` after the restore finishes, but before you try to restart the IS. (See the section "Using the `-patch` switch" for more details.)

©1999 Robichaux & Associates. All rights reserved. Visit http://www.robichaux.net

Third-Party Backup Tools

Many of the enhancements on the backup management front aren't particularly beneficial to Exchange administrators. For example, for normal Windows NT file backups, it may be desirable to have a web interface to the backup program allowing a user to search hundreds of thousands of files to specify which one to restore. One area where third-party backup tools really look good is in hardware support. Many products support auto-loading tape drives, multiple tape drives configured in striped arrays, or backup devices that *ntbackup* doesn't support. Apart from hardware support, many larger organizations have chosen backup solutions that can handle multiple operating systems or special needs like hierarchical storage management.

While third-party products can often do things that *ntbackup* can't, many of them have a history of problems with their Exchange support interfaces. Microsoft has provided a set of API routines that third-party products can use to scan the IS, but not every vendor has been able to successfully decipher Microsoft's documentation and build a usable backup tool.

Most third-party products that advertise Exchange support are usable, provided that you're careful. Make sure you get the correct agent or plugin module to properly back up Exchange, and make sure that the version you're using is the right one for the combination of Exchange, NT, and service packs that you're running. Spend some time researching the msexchange list archives, the Microsoft KB, and your peer network to find out whether other sites with similar configurations have had good luck with the solution you want to use.

No matter what else you do, *always* test your backup and restore setup on a complete copy of your database (with a recovery server, of course). It's imperative that you find problems with your backup software or hardware in the controlled environment of your test lab, not during a real service outage.

Maintenance & Repair Tools

As discussed in Chapter 11, Exchange can perform many database maintenance tasks either on its own or according to a schedule you set. These tasks include tombstone cleanup, online defragmentation, index expiration and aging, and other tasks associated with regular preventative database maintenance. However, there are times when offline tools are required for database compaction, testing, and repair.

©1999 Robichaux & Associates. All rights reserved. Visit http://www.robichaux.net

> Don't use these tools unless absolutely necessary. Like firearms, they're irreplaceable when you really need them, but they can be dangerous in careless or untrained hands.

eseutil

eseutil is a command-line program that performs a variety of functions, including compacting, testing and repair , on the database. It operates on individual 4KB database pages, not on messages, mailboxes, or folders. Exchange 5.5's version of *eseutil* can check database integrity at about 10GB/hourand repair them at 8-10GB/hour; *isinteg* can defragment databases at 4-5 GB/hour.

> *eseutil* replaces *edbutil*, the utility program used on Exchange 4.0 and 5.0 databases. In Exchange 5.5 SP1 and later, it's moved to the *winnt\system32* directory instead of *exchsrvr\bin*.

eseutil works in six distinct modes. For most Exchange systems, the only modes you'll be interested in are the defragmentation, integrity check, and repair modes. Each of the modes is discussed its own section below, so I'll leave the individual mode descriptions there.

When to use it

Ideally, you'd never run *eseutil*.I always cringe when I see people running it as a preventative maintenance tool. This is somewhat like doing preventative maintenance on your car with a welding torch--it gets the gunk off, but one wrong move and your engine will be a melted lump of slag. There are only three circumstances when I recommend running it:

- When you want to check the integrity of a database, either *in situ* or from a backup.
- When you need to defragment a database because you need the disk space. For example, if you move several dozen mailboxes to another server you can reclaim their space by an offline defrag. Don't, however, do these routinely; there's no reason to do so since the online defrag process runs daily.
- When you need to fix a corrupted database because you can't restore it from a backup, or because Microsoft tells you to.

I can't overemphasize that this is not a tool for casual or everyday use. It can be dangerous, especially in repair mode.

©1999 Robichaux & Associates. All rights reserved. Visit http://www.robichaux.net

How to use it

The first thing you have to cope with are the mode switches that control what *eseutil* does, as shown in Table 17-x.

Table 17-x. ESEUTIL command-line switches

Switch	What it does
/D	Defragmentation mode: copies the specified database to a new file, then defragments the file to make its data contiguous. When defragmentation finishes, copies the new file back to the original location.
/G	Integrity check mode: validates checksum and header information against the actual database contents. Nondestructive.
/M	File dump mode: dump the database file's contents in (mostly) human-readable form.
/P	Repair mode: validates the database table structure and links, truncating or changing things where necessary. May cause data loss. Use as a last resort.
/R	Recovery mode: attempts to put databases in a consistent state by repairing bad table links, but doesn't truncate or otherwise modify data in the tables.
/U	Upgrade mode: rarely used, since it's designed to update an older database schema to the current revision. Normally Exchange's setup and service pack installation programs do this.

Here's a complete breakdown of *eseutil*'s options:

```
eseutil
            | /D database [/L logPath] [/S systemPath] [/B backupName]
                  [/T tempName] [/P] [/O]
            | /G database [/T tempName] [/V] [/X] [/O]
            | /M[mode] fileName
            | /P database [/T tempName] [/D] [/V] [/X] [/O]
            | /R { /IS | /DS } [/L logPath] [/S systemPath] [/O]
            | /U database /D dllPath [/B backupName] [/T tempName] [/P] [/O]
```

Defragmentation mode

Exchange normally defragments the IS databases while the IS runs. However, you can do an offline defrag with *eseutil*; since the services aren't running, the utility can do a better job of compacting the database. Microsoft recommends that you do a full online backup after doing an offline defrag; that's because any outstanding log files will have the wrong

©1999 Robichaux & Associates. All rights reserved. Visit http://www.robichaux.net

database signature after the defrag finishes. In my opinion, you should do oen before the defrag, too, just in case something goes wrong.

Defragmentation mode has its own set of switches:

```
eseutil /D database [/L logPath] [/S systemPath] [/B backupName]
                    [/T tempName] [/P] [/O]
```

database
: Specifies what database you want to defragment. Use `/ds`, `/ispub`, or `/ispriv` as the database name to tell *eseutil* to look up the database name and path in the registry, or provide the full path and database name.

logPath
: Specifies where the transaction log files for this database are. Defaults to the current directory if not specified; not required when using the `/ds`, `/ispub`, or `/ispriv` switches.

systemPath
: Tells *eseutil* where to find the checkpoint file. Defaults to the current directory.

backupName
: Forces *eseutil* to make a backup of the database being worked on, using the specified name and path.

tempName
: Specifies a name for the temporary database that eseutil creates. Useful for redirecting the temporary database to another disk where you have more space. Defaults to *tempdfrg.edb* in the current working directory.

The `/p` option tells *eseutil* to preserve the temporary database, so it will create it but not replace the original with the newly created file. You would then need to manually replace the original file with the newly created temporary file. Finally, the `/o` option just suppresses eseutil's version and copyright message.

Integrity check mode

eseutil can verify the low-level integrity of the database and its pages. Note that this isn't the same as what *isinteg* does; that utility checks the integrity of message and maiblox items in the database. This mode is nondestructive, but it assumes that the database is in a consistent state when you run it; if not, you'll get an error.

```
eseutil /G database [/T tempName] [/V] [/X] [/O]
```

database
: Specifies what database you want to defragment. Use `/ds`, `/ispub`, or `/ispriv` as the database name to tell *eseutil* to look up the database name and path in the registry, or provide the full path and database name.

/V
: Turns on verbose mode, which provides a wealth of information about what the utility's doing.

©1999 Robichaux & Associates. All rights reserved. Visit http://www.robichaux.net

/X
: Forces eseutil to provide detailed error messages instead of its usual terse ones.

The `/T` switch has the same function as in the defrag mode; it specifies where to store the temporary file.

Dump mode

The dump mode just tells eseutil to print some information about either the database header or the checkpoint file. It's mostly useful if you're curious about what's in those files or if you're asked to dump the files during a call to Microsoft support.

```
eseutil /M[mode] fileName
```

mode
: Specifies what dump mode you want to use. The `K` modifier specifies a checkpoint dump, and the `H` modifier (the default value) specifies a header dump.

filename
: Specifies the full path and filename of the file whose contents you want to see.

Repair mode

eseutil can attempt to repair damaged databases; it does so by checking the database's links between various tables of information and fixing those links if it can tell that they're bad. This repair operation is nondestructive, but it's not guaranteed to return the database to a consistent state when you run it.

```
eseutil /P database [/T tempName] [/D] [/V] [/X] [/O]
```

database
: Specifies what database you want to repair. Use `/ds`, `/ispub`, or `/ispriv` as the database name to tell *eseutil* to look up the database name and path in the registry, or provide the full path and database name.

/D
: Specifies that eseutil should test the database for errors without repairing it.

The `/T`, `/V`, `/X`, and `/O` switches have the same function here as in the previous modes.

Recovery mode

This mode is scary because it can cause data loss. When you tell *eseutil* to recover a database, it will freely truncate any database page it can't cleanly recover. While this will normally restore your database to a consistent and usable state, it will also normally cause you to lose some message and/or mailbox data. Don't use this mode except as a last resort. If you run an integrity check and it shows errors, always run a repair first. If that doesn't fix everything, you have two choices: restore from a good backup (hopefully with no data loss), or run a recovery. Any time you're tempted to choose the latter option, call Microsoft support *first* to see whether there are any other alternatives for recovery. MS

©1999 Robichaux & Associates. All rights reserved. Visit http://www.robichaux.net

has an array of specialized tools that they can give you to fix specific problems, but only if you call them.

```
eseutil /R { /IS | /DS } [/L logPath] [/S systemPath] [/O]
```

The interesting switch here is the one that controls whether the recovery runs against the IS or DS. You can specify either, but not both, and *eseutil* will automatically look up the location of the log and database files in the registry—you can't manually override those values. The `/L`, `/S`, and `/O` switches work the same way here as in the other modes.

Update mode

This mode is rarely used. In fact, Microsoft's documentation says its use will usually only be required "with the release of a major, new revision of Microsoft Exchange Server."

```
eseutil /U database /D dllPath [/B backupName] [/T tempName] [/P] [/O]
```

`database`
: Specifies what database you want to upgrade. You have to give the full path and database name; there aren't any shortcut switches.

`dllPath`
: Specifies the full path to the database DLL for the version of Exchange you're upgrading from.

The `/B`, `/T`, `/P`, and `/O` switches work the same as in other modes.

isinteg

This utility does two things:

- It can test the IS databases for logical errors and fix them. In this mode, it verifies the integrity of information in the database, not of the database itself (that's eseutil's job). To do this, it cross-checks information in about 20 tables to determine what state the database is in. More specifically, it searches the IS databases for table errors, incorrect reference counts, and orphaned objects, none of which should exist in a consistent database.
- It can patch the IS after you restore it from an offline backup. This is necessary because restoring an offline backup of the IS databases doesn't restore some internal fields of the database, but the patch mode will.

Microsoft recommends against using *isinteg* to fix database errors unless they tell you to. As with *eseutil*, there's some risk involved with running *isinteg*; however, I think it's reasonably safe to run it if you know what you're doing. However, be forewarned that running it may cause data loss- don't do it unless it's necessary.

©1999 Robichaux & Associates. All rights reserved. Visit http://www.robichaux.net

When to use it

The most common use for *isinteg* is to patch the store after running an online backup, as discussed in the "Using the –patch switch" section. Apart from that, any time the IS won't start, you should run *isinteg* in test mode so it can check the IS for errors. This is particularly true if you see IS errors in the event log. There are other circumstances when you might suspect that something's amiss:

- Inconsistent message count on private or public folders. For example, a folder may show 5 new messages when only 3 exist. *isinteg*'s reference count tests are used to address such issues.
- Unexplained crash of information store when a user access a given folder or message.
- User unable to access a message or folder from any client due to client error. Event log entries may also be present on server containing messages.

It never hurts to run *isinteg* in test mode; however, you should only run it with the `-fix` switch if you've got a recent backup.

How to use it

The full set of *isinteg*'s options look like this; they're explained in Table 17-x.

```
isinteg [-pri] [-pub] [-fix] [-L [logFile]] [-detailed]
    [-verbose] [-test { alltests | testName} ] [-dump] [-[patch]
```

Table 17-x. ISINTEG command-line switches

Switch	What it does
`-detailed`	Provide additional detail on any database problems found
`-dump`	Verbose dump of store data. Interesting, but not always useful.
`-fix`	Fixes problems found during the integrity check. Without this switch, a read-only check is performed.
`-L`	Specifies the name of the *isinteg* log file. Defaults to *isinteg.pub* or *isinteg.priv* in default directory.
`-patch`	Patch information store after an offline restore. (See the section "Using the –patch switch" below).
`-pri`	Specifies that isinteg should check the private information store, *priv.edb* (it gets the file's location from the registry)
`-pub`	Specifies that isinteg should check the public IS, *pub.edb*
`-T`	Specify path to database files; normally extracted from

©1999 Robichaux & Associates. All rights reserved. Visit http://www.robichaux.net

	registry.
-test	Specify which tests will be performed. -test alltests is recommended, since it runs all tests in sequence. You can also name individual tests. Following tests are named: Folder/message tests: folder, message, aclitem, delfld, acllist, timedev, rowcounts, attach, morefld, global, searchq, dlvrto, search, dumpsterprops, namedprop Private IS only: rcvfld , mailbox, oofhist Public IS only: peruser, artidx, newsfeed Reference count tests: msgref, msgsoftref, attachref, acllistref, aclitemref, newsfeedref(pub only), fldrcv(pri only) fldsub, dumpsterref Groups tests: allfoldertests, allacltests Special tests: deleteextracolumns
-verbose	Provide verbose progress messages

Using the -patch switch

isinteg is also used to patch the database when you restore it from an offline backup. That's necessary because of the way the IS allocated object IDs. Each object in the public and private IS databases have a globally unique identifier, or GUID. Object GUIDs are derived from the base GUID of the store they live in. Microsoft uses GUIDs in Exchange to uniquely identify an object's location *and creation time*. When you do an offline restore, you're reloading an "old" version of the database: in effect, you're turning back time. If you don't change the store's base GUIDs, newly created objects could accidentally get GUIDs that match an item already in the store, and that would cause major trouble for replication.

When you do an online backup,*ntbackup* fixes the GUIDs as it does the restore; for an offline backup, you must manually fix them by running *isinteg* -patch. If you don't run it after doing an offline restore, the IS won't start, and it will record error –1011 in the event log. The message for that event says (paraphrased) "You restored an offline backup. Go run isinteg –patch or I won't start."

To use the -patch switch, make sure that the DS and SA services are running, then run *isinteg* -patch from the command line. It will replace the GUIDs, after which you can safely restart the IS. Note that you can't patch one IS or the other; *isinteg* will always patch both databases.

©1999 Robichaux & Associates. All rights reserved. Visit http://www.robichaux.net

Recovering Your Data

Chapter 15 covers basic troubleshooting of the Exchange Server environment; it is a good reference point to start from. Once you've determined what's wrong, how do you fix it? Understanding how to fix specific problems is useful only if you can match your specific problem to the corresponding solution. Table 17-x summarizes common problems and their solutions; the rest of this chapter will discuss the solution steps in detail.

Table 17-x. Troubleshooting guide

Problem	Server condition	Procedure
Directory database (*dir.edb*) damaged or missing	· NT installation is OK. · Exchange installation is OK. · The server's directory is corrupted or unavailable (including disk failures)	Restore the directory database from a known good backup (see the section "foo")
Public and/or private IS database damaged, logs OK	· Windows NT and Exchange are undamaged · The *priv.edb* and/*or pub.edb* files are lost or damaged. · Circular logging is off. · You have a usable backup.	Restore the IS and log files to the failed server. Restart the IS to force it to play back the log files. See the section "bar".
Private IS damaged or lost, no logs	· Windows NT and Exchange are undamaged · The *priv.edb* and/or *pub.edb* files are lost or damaged. · Full logs are unavailable (log drive has failed, backup is bad, or circular logging was on). · You have a usable backup of the IS.	Prepare for data loss! Restore from most recent full backup, then restart IS to play back any remaining log files. Some changes made since full backup will be lost.
Public IS damaged or lost, no logs	4. Windows NT and Exchange are undamaged 5. Replicas of the public folders exist	Restore the old public IS database, then allow public folder replication to bring the contents up to date
Single mailbox deleted by human error	2. Server is undamaged. 3. No OST is available.	Create a new mailbox for the user on the production server, then see the section "Recovering Data From

©1999 Robichaux & Associates. All rights reserved. Visit http://www.robichaux.net

		One Mailbox".

One of the more challenging aspects of dealing with recovery is the combination of losing Windows NT and Exchange Server at the same time. Not just an Exchange Server database corrupting, but the inability to boot or recover the NT components of a system.

There are basically only a few considerations:

- Is the hardware working, or was it the cause of the failure? If the hardware is 'safe', and the cause of the problem was human error, software, or some other factor that was removed. In simple terms, are you re-using the old hardware or do you need to spend time acquiring new or spare hardware?
- Can the data be recovered from the system? If the database files can be copied off, you may wish to bypass any tape restoration solutions and jump right to a offline restoration. If the system won't boot to access NTFS, one technique to get to the EDB files is to boot up from a Windows NT CD-ROM and install a second copy of NT on another partition to get on there long enough to copy the files off. Of course, options here depend on the specifics of the failure and the disk layout. I've also seen drives moved off of one computer and connected to a controller of a different computer. Get your NT experts, as this is just a simple file retrieval—when the Exchange services aren't running, these files are just like any other files on a Windows NT system.
- How long do you have? If rebuilding a server will take time, do you have another suitable machine already running that could either join the domain or be renamed in the domain?

Recovering Data From One Mailbox

The largest surprise to most new Exchange Server Administrators is the lack of ability to easily restore a single mailbox, folder, or message. In fact, the Exchange design requires that an entire *priv.edb* or *pub.edb*file be restored all at once. Practically, this means that the entire server's mailbox contents must be restored to retrieve a single user's deleted mailbox.

To recover a single message in a single mailbox or public folder, you have to restore the entire private or public IS. That's a huge amount of effort. Exchange 5.5's deleted item recovery feature means that most of the time you can get away without having to go through the whole process, so I recommend that you turn it on and give it a liberal retention period. (Note that it won't help you if the client's using PSTs, POP3, or IMAP4.)

What if you need to recover an item that's been removed but isn't in the dumpster? You can't restore the IS to your production server, because that'll overwrite changes to

©1999 Robichaux & Associates. All rights reserved. Visit http://www.robichaux.net

everyone else's mailbox. Instead, you need a separate recovery server. That server will come in handy for other recoveries, too, so if at all possible you should keep one handy.

The recovery server

Your recovery server needs to have enough disk space to install NT and Exchange (with all the service packs you use on your production servers), plus enough space to restore the private IS. It can be on the same LAN as your production network; however, if you leave the recovery server up you must be sure that it doesn't try to participate in directory replication.

Before you can recover anything, you have to prepare the server appropriately:

* Install Windows NT and any service packs. This computer can be a PDC, BDC, or member server, and its network name is unimportant because you won't be restoring the DS, just the IS.
* Install Exchange. When prompted, create a new site, using the organization and site name from the server whose backup you're restoring. *Don't join the existing site*.
* Install whatever Exchange service pack was installed on the server at the time of the backup. For example, if you're restoring from a backup made while SP1 was installed, install SP1 on the recovery server even if you've since upgraded your production machine to SP2.
* Install the Exchange or Outlook client on the recovery server.

You may or may not need to repeat these steps in the future; if you have a dedicated recovery server, you may be able to leave it alone once it's set up, or you may have to reinstall Exchange or even NT to match the configuration of the server you're trying to restore from.

Recovering a single item

Once your recovery server is up and going, the actual recovery is straightforward. The following steps assume that you've logged on to the recovery server as an administrator.

· Restore the IS to your recovery server, either from an offline or online backup. Make sure the IS starts when you're done.

· Run Exchange Administrator, then start the DS/IS consistency adjuster. (It's in the Advanced tab of the server's properties dialog). This is required to populate the directory, since you didn't restore it.

· Open the recipients container, find the mailbox you want to restore, and open its properties dialog. Use the Primary Windows NT Account button to select the account you logged on with as the mailbox owner.

©1999 Robichaux & Associates. All rights reserved. Visit http://www.robichaux.net

- Configure a messaging profile for the new user account, making sure to add the Exchange and Personal Folder services to it.
- Launch the Outlook or Exchange client. If you're using the Exchange client, select the user's folder and copy it, then paste it to the Personal Folders item. If you're using Outlook, use the File-->Export… command to export the desired data to a PST.
- Deliver the PST to the original client, or move the PST contents into their mailbox for them. Warn them sternly not to lose any more data.

Of course, these steps work whether you want to restore an individual message or an entire mailbox.

Restoring Data When Your Machine Is OK

Restoration is where backup solutions test their mettle. You are strongly advised to test your backup solutions as much as you can tolerate, as there is no point in doing a backup if the restore is unsuccessful. All too often, automated backup solutions get out of hand and errors are not caught until it is too late.

The act of restoring can present a risk itself. Accidentally restoring over an active server can have disastrous results. In fact, Microsoft's documentation advocate a dedicated recovery server as a primary means of recovery. The following excerpt from the documentation:

> When a mailbox or information store is corrupted, you can use backups to recover the information store to a dedicated recovery server and then restore the mailbox or information store to the production server. When a server fails, you can use backups to restore the server's information store, directory, and configuration to a recovery server and then place the recovery server in production to replace the failed server.

Before attempting any restore or recovery efforts, make copies of all the existing LOG files. it is also advised to take the time and copy the existing database and log files. in fact, it may be simplest to stop all of the exchange server programs, then XCOPY the various *\exchsrvr* file trees (which may be on multiple partitions). With large stores, this could take some time, but could be of considerable use if the restoration is problematic or in determining the cause of the failure.

Restoring from a failed database drive

This is probably the easiest type of recovery, because if you have a good backup and a good copy of the log files Exchange can cleanly repair itself. If you lose the public or private IS or directory databases because the drive they're on fails, here's what to do:

* Use the Services control panel to disable and stop the SA service.

©1999 Robichaux & Associates. All rights reserved. Visit http://www.robichaux.net

* Replace the failed drive. Create a new logical drive with the same name as before, then format it.
* Create a directory structure for Exchange identical to the one on the failed drive. (If you need to, you can cheat and look in the registry to get the correct structure.) Normally, this means you need to create the *exchsrvr* directory with subdirectories named *mdbdata* and *dsadata*.
* Restore the databases from your last backup. If possible, use an online backup. Don't worry about restoring the transaction logs (if they were on the same drive, you'll need to follow the steps in the next section).
* Enable the SA service, then start the SA, DS, and IS services. When the IS starts, it will replay the transaction logs and bring the restored IS or DS up to date.
* Check the event log to make sure everything went smoothly.

Restoring from a failed log drive

When your log disk fails, you're probably going to end up losing some data unless your most recent backup is very recent indeed, although you may be able to recover some data using a separate procedure that I'll get to in a minute. First, here's what you need to do when you lose the disk with the IS logs on it:

1. Use the Services control panel to disable and stop the SA service.
2. Replace the failed drive, then create a new logical drive with the same name as before.
3. Format the new disk and create a directory structure for Exchange identical to the one on the failed drive. In particular, you need the *exchsrvr\mdbdata* directory.
4. Back up the IS databases, either online or offline.
5. Restore the most recent online backup of your IS databases.
6. Enable the SA service, then start the SA, DS, and IS services. When the IS starts, it will contain only the data from the time of the last backup.
7. Check the event log to make sure everything went smoothly.

When you lose the DS log disk, you're in better shape because the directory can repair itself via replication. All you have to do in that case is fix the broken disk (following steps 1-3 above), back up your existing directory, then restore the most recent online backup of *dir.edb*. Once you do that, and restart the SA and DS services, the normal replication process will backfill any missing data.

©1999 Robichaux & Associates. All rights reserved. Visit http://www.robichaux.net

If you want to try your luck at extracting additional data from the IS itself, you can. Some data that would normally be available from the logs might be available in the IS, in which case you can retrieve it by using the consistency adjuster. However, you *must* attempt this on the recovery server, not your production machine.

1. Set up your recovery server using the instructions in the section "The recovery server" earlier in this chapter. As in that section, be sure to use the same organization and site names as the production server, but *don't* join an existing site—create a new one with the same name.
2. Make a backup copy of all files in the *exchsrvr\mdbdata* directories on your server.
3. Copy the private and public IS databases from the production server to the recovery server.
4. If necessary, use *isinteg* and/or *eseutil* to repair the databases. Once the databases are consistent, start the IS.
5. Run the DS/IS consistency adjuster. This may make changes to your IS and directory, which is what you're hoping for.
6. Use the *exmerge* tool (covered in Appendix A) to merge whatever data the adjuster adjusted from the recovery machine back to your production server.

Restoring Exchange When NT Is Damaged

As long as you can accurately recreate the underlying NT configuration of your Exchange server, restoring to it is not significantly harder than restoring when a disk fails. Restoring the IS is exactly the same, in fact; the difference is that you normally need to restore the DS as well, and that's a little more complicated. There are two new requirements that you must be able to meet to successfully restore the directory:

- The recovery server must have the same organization, site, and server name as the original machine.
- You must have access to the original domain's SAM database.

Of course, the server you're restoring onto has to have adequate capacity to hold the databases, enough disk space to install Exchange and Windows NT, and so on. Be careful to accurately replicate the server's original NT configuration—install the same hotfixes, service packs, and third-party services in the same order as the original installation.

Restoring domain controllers

You have to be especially careful during recovery if you've installed Exchange on a domain controller. That's because having access to the domain SAM is a prerequisite for a

©1999 Robichaux & Associates. All rights reserved. Visit http://www.robichaux.net

successful recovery. If your failed machine was a BDC, you're in good shape—as long as the PDC or another BDC is still on the network, you can reinstall NT on your Exchange server as a BDC, and you'll be OK. The same is true if the server you were restoring was a member server.

The sticky part comes when you try to restore Exchange onto a machine that was formerly a PDC. If you reinstall NT as a PDC, it will create a totally new SAM database that won't match the original; although you'll be able to restore the IS and DS from a backup, you won't be able to start the Exchange services or use the restored directory, since it depends on the original security context. Without a directory, you can't do anything except restore individual mailboxes.

The key is to make sure that you have an available PDC when you do the restore. It may be a BDC that you've promoted (in which case you make your newly-reinstalled server a BDC and promote it later), or it may be the original PDC. As long as one domain controller is available, you won't have any problem.

Restoring Exchange

The actual steps involved in restoring Exchange to a server that's been reloaded with a fresh copy of NT are as follows:

1. Remove the computer's old domain account on the PDC/BDC, then add it back.
2. Log on to the target machine as a domain administrator.
3. Run Exchange Setup using the `/r` switch. (See Chapter 4 for details).
4. Make sure the server name matches the original server name (it should, as long as the NT names are the same).
5. Create a new site using the exact same site and organization name as the original server. Upper and lower-case letters are different to Exchange, so make sure you have the capitalization right.
6. When prompted, use the same service account as the original server.
7. Install the same connectors that were on the original server.
8. Install the same Exchange service pack as was on the original server.
9. Configure the IMS, INS, MS Mail connector, and any third-party connectors , since they may store their configuration parameters in the registry instead of the directory.
10. Run Performance Optimizer.
11. If KMS was installed on the original machine, reinstall it.

©1999 Robichaux & Associates. All rights reserved. Visit http://www.robichaux.net

12. Install Outlook or the Exchange client.

Don't start the Exchange services. At this point, you've got a fresh installation of Windows NT and Exchange, but you still have to reload your data from your backups.

Restoring your data

The restoration procedure varies slightly, depending on whether you have an online or offline backup and whether you have any log files generated after the original backup. I'll note the differences where they occur. Here's how to restore your data:

1. If you have transaction logs generated after the original backup, copy them to the log directories of the recovery server.
2. If you have an online backup, restore it using *ntbackup*. Tell *ntbackup* to back up the private and public IS, turn on the "Start Services After Restore" checkbox, and make sure the "Erase all Existing Data" box is checked" *unless* you have transaction logs from after the original backup.
3. If you have an offline backup, stop the Exchange services on the recovery server and copy the database and log files to their proper locations, then restart the DS and SA and run *isinteg* `-patch`. Once that finishes, restart the IS.
4. If you're running KMS, stop the KM service and restore its data, then restart it.
5. Verify mailbox account associations by opening a mailbox's properties dialog and checking the Primary Windows NT Account field. If you used the correct domain SAM, you should see that the account is correct.
6. Use the client software you installed in step 11 above to make sure that you can log on as a user, see calendar data, and exchange mail with other users.
7. Repeat the previous step using someone else's workstation.
8. Reconfigure Exchange and NT to match the original configuration. In particular:
   - Increase the size of the application event log
   - Make sure the pagefile is set to the correct size
   - Turn off circular logging
   - Add any alternate service accounts
   - Set diagnostic logging levels, INS, and IMS settings as desired—they're stored n the registry, not the directory.

©1999 Robichaux & Associates. All rights reserved. Visit http://www.robichaux.net

Restoring Exchange To A Different Computer

There are circumstances where you may be willing to lose configuration data but are more concerned with mailbox (*priv.edb*) content. As long as you keep the server name the same (and get a new domain account for that name), you can restore Exchange to a new server—but what if you want to move it somewhere else? Perhaps your Exchange installation is damaged, but the server is still able to run other services and you are unwilling to rebuild the entire system—instead, you want to move Exchange to another system with a new name.

This is not an ideal solution, as configuration data will be lost, all clients will need to be told about the new server, and so forth. It is best suited for testing of restore procedures, single mailbox restore, and other data-only restores (where clients and other servers never directly connect to this restored data).

One variation of an offline restore is an alternate server restore. The normal Exchange restore procedure doesn't allow you to create a *priv.edb* or *pub.edb* file directly. To accomplish this, you would need to restore your database to an alternate server, stop that server, and manually copy the file you desire. If you do this, the server name and other aspects of the server won't be identical to the one you intend to deploy the store database on.

The fastest way to do this is to do a restore as outlined above. When you try to restart the IS, Exchange will complain (via event log entries, including event ID 143) that the logs and the databases don't match. Your only possible response to that is to remove the log files, then restart the IS. This costs you the data in the log files; whether that's acceptable or not depends on what you're trying to do (be sure not to do it until you have a good backup of the log files!)

Once you've done that, you still need to use the DS/IS consistency adjuster to bring the directory in line with what's in the mailboxes. Figure 17-x shows the DS/IS Consistency Adjustment screen. In this case, we want to synchronize the private IS and the directory, so the “Synchronize with the directory, and create new directory entries for mailboxes that do not have a corresponding directory entry” box is checked. It exactly describes our situation, as we used the restoration from tape to put the mailboxes in place without restoring the directory.

©1999 Robichaux & Associates. All rights reserved. Visit http://www.robichaux.net

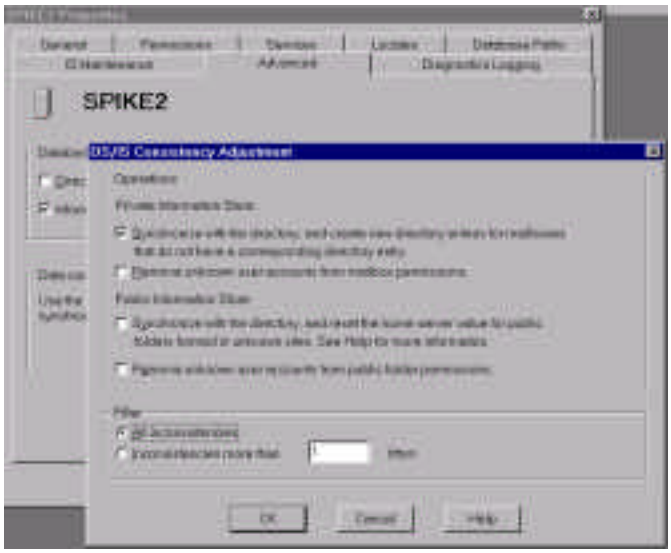

Figure 17-x. DS/IS Consistency Adjustment screen, options selected to discover users from alternate server priv.edb *file.*

One last step is to change the Filter setting to "All inconsistencies" from the default of "Inconsistencies more than 1 days". Press OK to start the adjustment. After the adjuster finishes running, the directory entries will now appear. You will notice that the directory entries are missing all their normal fields—most of the mailbox information is stored in the directory, and we didn't restore it. However, what did get restored is enough to allow you to recover data from the mailbox and print it, look it over, or whatever.

Directory Recovery

Complete recovery of the database is often not as much of an issue. Given relatively small file size, replication with other servers in the same site, most Directory issues are not related to total file loss or corruption.

Not to say that problems do not exist. Given how much the Directory is at the center of the server's behavior, it is critical to take care in changes or actions that impact the Directory. For specific recovery procedures, the Microsoft White Papers and Knowledge Base articles should be referenced. One particular Knowledge Base article Q196406, titled "XADM: Replication Fails After Disaster Recovery", actually covers much more than the title implies. It includes a further reference Q15960, titled "XADM: Rebuilding the Site Folders in a Site".

©1999 Robichaux & Associates. All rights reserved. Visit http://www.robichaux.net

Also check into Q162353, titled "XADM: Restoring an Exchange Directory". These articles outline a strategy to rebuild the directory. Although this article somewhat oversimplifies the operation, it does emphasize how basic tools such as file copy and directory import/export can be used when proper backup solutions fail. When no copy of the directory is available, data in the IS can still be recovered. A call to Microsoft Support or consultation with someone who has done this procedure is in order for such a situation, as there are several options and version-dependent concerns.

One of the most important steps is to quarantine the server. If new mail delivers or directory replication takes place, irreversible changes can be made. This can be taken to all servers in a site in some circumstances, carefully consider the nature of the damage.

Preventative Medicine

How can you ensure that your recovery goes smoothly? Practice makes perfect, but adequate documentation and good configuration control helps too. You should maintain the following items *and* audit them periodically to make sure that the documentation and the real-world implementation match:

- If you're using a TCP/IP environment, keep *hosts* and *lmhosts* files for all the Exchange servers, backup servers, domain controllers and other critical systems that the Exchange server system may require.
- Keep documentation on backup locations and procedures, including where off-site tapes are stored and how to get to them. Include details such as contact information for all persons and companies involved in these responsibilities.
- Export the entire directory to a CSV file every week or every night as a precaution.
- Make sure records of each server installation, including the service account username and password, are available. Of course, this information should be secured, but it is also important to ensure that the current service account password is available for server recovery.
- Make regular (at least weekly) registry backups. Be sure to make additional backups after hardware, operating system, or Exchange configuration changes.
- Verify the backups at least once after every Exchange Server service pack, version upgrade, or other major change.
- Put your databases and logs on separate physical disks.
- Turn off database circular logging.

©1999 Robichaux & Associates. All rights reserved. Visit http://www.robichaux.net

- Consider procedural issues when a mailbox is terminated. For example, hiding a mailbox for 30 days before deletion could prevent the need to restore the mailbox if an employee returns or replacement staff requires access to content.
- Have a basic plan on how to react if a database becomes corrupted or a system needs to be otherwise restored. For example, do you want to take the time to do a full file-by-file tape backup of the system *before* you start any attempts to recover or restore? This could be of use when later attempting to find out why the failure took place.
- Have a suitable null modem cable on hand at all times and familiarize yourself with use of the kernel debugger to trace system boot-up. This is valuable when you one day find that your NT server won't boot due to a device driver, hardware device that failed, or corrupt system file.
- Make sure you run *rdisk* after each configuration change. It may be best to do so weekly or daily as part of routine maintenance.

As I mentioned at the start of this chapter, when your backups are properly planned and implemented, the next challenge is to help cut the recovery time down to the minimum. You can do this by buying backup hardware that's sized appropriately to the task at hand; in some cases, you may want to consider splitting up large private or public IS databases by moving some of their data to another server. Chapter 14 discusses some of these issues and explains what solutions and practices make sense.

Offline Defragmentation

If you are concerned that the store may have minor problems, one way to regain assurance is to run an offline defragmentation or integrity check on your recovery server. This is also a good way to establish performance baselines for a particular system.

If you do decide to defragment your store, you must make a full *online* backup after the defrag finishes. When the offline defrag process runs, it changes the signature on the database, making it incompatible with older log files.

Reducing the Risk of Mailbox Loss

More precisely, what can you do to reduce the risk of losing mailbox data? Assuming you have good backups, and that you've turned on deleted item retention to guard against accidental deletions, what else can you do? There are several potential solutions, some of which are pretty creative:

©1999 Robichaux & Associates. All rights reserved. Visit http://www.robichaux.net

- Use scripts or an alternate mail client to copy the mailbox contents on a message-by-message basis. Exchange 5.5 does not support server to server replicas of a mailbox, but you can deliver this type of feature with some clever scripting.[1]
- For mailboxes with critical data, consider offering users their own public folder for longer-term storage. Most users create various folders for storage based on projects, work areas, and other organizational methods. By leveraging the Outlook shortcut bar and the public folder favorites category, you can make it easy for them to use a tree of individual public folders instead of mailbox folders. Part of this solution involves overcoming the literal meaning of "public folder" by establishing a good tree structure and proper permissions to make these as secure as the mailbox. One big advantage of this solution is that public folders can be replicated, limiting the bulk of the mailbox and simplifying backup and recovery.
- Use the alternate recipient features to tie the mailbox to a second mailbox on another server or a public folder. This provides a backup copy of the mailbox contents; one trick is to create a public folder, set the permissions to default write, unhide the folder from the Global Address List, then put that Public Folder in as an alternate recipient.
- Using Outlook's ability to clone the user's mailbox contents with an OST. With a little scripting, you can establish a way to automate opening the mailbox, selecting all folders, synchronizing them all, then copying the OST to safe location. This way just the OST can be restored.
- Using a third-party product that supports mailbox-by-mailbox, or "brick-level", backup. These backup applications actually open each mailbox and download individual items. The restore process involves an automated fetching of the mail and uploading it back into the mailbox. They're slow, and they often don't work very well, but they may be worth investigating if none of the other alternatives meet your needs.

---

[1] Especially considering that the Win32 version of Perl supports MAPI!