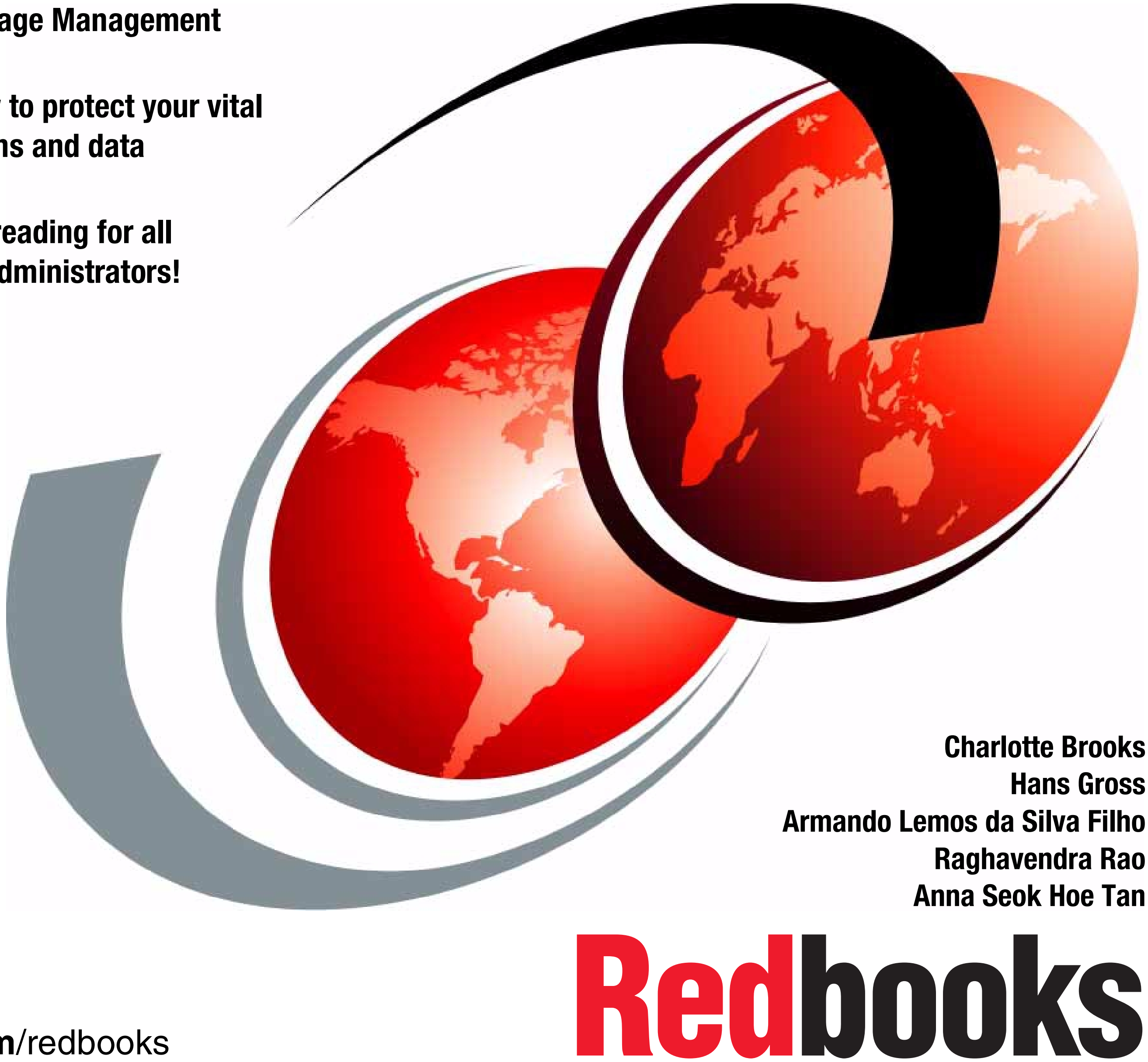
Tivoli

IBM

Tivoli Storage Management Concepts



Ideal introduction to all aspects of Tivoli Storage Management

Learn how to protect your vital applications and data

Essential reading for all systems administrators!

Charlotte Brooks
Hans Gross
Armando Lemos da Silva Filho
Raghavendra Rao
Anna Seok Hoe Tan

Redbooks

ibm.com/redbooks

International Technical Support Organization

SG24-4877-02

Tivoli Storage Management Concepts

August 2000

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix A, “Special notices” on page 291.

Third Edition (August 2000)

This edition applies to Version 3 Release 7 of Tivoli Storage Manager for AIX, Solaris, HP-UX and NT, program 5697-TSM; Version 3 Release 1 of Tivoli Storage Manager for AS/400, program 5697-TSM; Version 3 Release 7 of Tivoli Storage Manager for MVS-OS/390, program 5697-TSO; and Version 3 Release 1 of Tivoli Storage Manager for VM, program 5697-TSO as well as complementary Tivoli Storage Management Solutions.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. 471F Building 80-E2
650 Harry Road
San Jose, California 95120-6099

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Figures

Tables

Preface

This IBM Redbook describes the features and functions of Tivoli Storage Manager. It introduces Tivoli Storage Management concepts for those new to storage management, in general, and to Tivoli Storage Manager, in particular.

This redbook gives a broad understanding of the architecture of Tivoli Storage Manager and will help you design solutions to protect your data holdings from losses ranging in scale from those caused by user error through complete site disasters.

Where specific "how-to" information is available, pointers are given to the locations of this information both in redbook form (if available) and in Web pages. These pointers will be given at the beginning of each topic. This redbook and the "how-to" information it points to will be continually updated as Tivoli Storage Manager evolves. You should check the online material for the latest information.

A companion redbook is available, *Getting Started with Tivoli Storage Manager: Implementation Guide*, SG24-5416, which addresses the practical side of planning, implementing and maintaining a Tivoli Storage Manager environment.

Tivoli Storage Manager is the follow-on product to IBM’s ADSTAR Distributed Storage Manager (ADSM). It is a part of Tivoli Storage Management Solutions. For more information, please visit the Tivoli Storage Management Web site at:

`http://www.tivoli.com/products/solutions/storage`

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization San Jose Center.

Charlotte Brooks is a Tivoli Storage Management Project Manager at the International Technical Support Organization, San Jose Center. She has ten years of experience with IBM as a technical specialist in the fields of RS/6000 and Storage Management. Before joining the ITSO this year, Charlotte was Technical Support Manager for Tivoli Storage Manager in the Asia Pacific region.

Hans Gross is an IT Specialist for AS/400 in Germany at the IBM Integrated Technology Services. He has over ten years of experience in AS/400 networking, transition management, problem determination and resolution as well as pre-sales support. For the last three years he has provided BRMS/400 and Tivoli Storage Manager/400 services and implementations for customers.

Armando Lemos da Silva Filho is the Information Security Officer at Deutsche Bank Brazil, UNIX Level 2 support and also an IBM certified specialist in Tivoli Storage Manager. He has 12 years of experience in the data processing field for Mainframe and Midrange systems. He has been working with Tivoli Storage Manager, AIX, and NT for the past four years. He holds a degree in Computer Science from Pontificia Universidade Catolica in Sao Paulo. His areas of expertise include Hypertext Technology, Computer Security, Operating Systems and Disaster Recovery.

Raghavendra Rao is a Country wide Software Support Professional from Integrated Technology Services - IBM India. He has over four years of experience which includes Lotus Domino, Tivoli Storage Manager and CATIA. He is also a Principal Certified Lotus Professional. His areas of expertise include infrastructure design, deployment, technical support, and application development for Lotus Domino/ Notes, Tivoli Storage Manager and Digital Library.

Anna Seok Hoe Tan is a software specialist with IBM Singapore's RS/6000 Services team. She has a degree in Computer and Communication Engineering from Essex University in England. She has 10 years of experience in IT in Asia, mainly in system integration. In IBM she provides customer consultation, project implementation and services, specializing in IBM RS/6000 based products including Tivoli Storage Manager and OnDemand.

The authors of the first edition of this redbook are:

Patrick Randall, a Distributed Storage Software Specialist at the International Technical Support Organization, San Jose, California.

Arnold Balingit, an IT Specialist in IBM Philippines.

Ross Battaglia, the Director of Technical Services at ETI, in Boca Raton, Florida. ETI is an IBM Business Partner.

J.P. Houle, a Senior IT Specialist with IBM Canada in Ottawa, Ontario.

Mathis Landzettel, a former Project Leader at the International Technical Support Organization, San Jose Center.

Roland Leins, a Project Manager at the International Technical Support Organization, San Jose Center.

Armando Lemos da Silva Filho, of Deutsche Bank, Sao Paulo, Brazil.

Rod MacLeod, an MVS Team Leader for TELUS Communications in Alberta, Canada.

Andy Pattinson, a Storage Consultant for Storm UK, an IBM Business Partner and European storage distributor.

Phil Thomas, a senior Technical consultant for Aspect Computing, an IBM Business Partner in Australia.

Thanks to the following people for their invaluable contributions to this project:

Michael Pousson
Tivoli Storage Management Marketing, San Jose

Yvonne Lyon
International Technical Support Organization, San Jose Center

Emma Jacobs
International Technical Support Organization, San Jose Center

Comments welcome

Your comments are important to us!

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Fax the evaluation form found in “ITSO redbook evaluation” on page 301 to the fax number shown on the form.
- Use the online evaluation form found at **ibm.com**/redbooks
- Send your comments in an Internet note to redbook@us.ibm.com

Part 1. Storage management concepts

The most recent version of this material is maintained on the ITSO Internet Web pages at:

`http://www.redbooks.ibm.com/solutions/systemsmanagement/tsm/concepts.html`

Part 1 introduces basic storage management concepts as they relate to Tivoli Storage Manager.

We cover:

- Distributed data management and Tivoli Storage Manager
- Tivoli Storage Manager architecture
- Data and storage management paradigms of Tivoli Storage Manager
- Additional major features of Tivoli Storage Manager

Chapter 1. Introduction to Tivoli Storage Manager

Tivoli Storage Manager is the core product of the Tivoli Storage Management product set. It provides a solution for distributed data and storage management in an enterprise network environment. Tivoli Storage Manager supports a wide variety of platforms for mobile, small and large systems, and delivers together with complementary products, many data management functions, such as data protection for file and application data, record retention, space management, and disaster recovery.

This chapter gives a high-level technical introduction to Tivoli Storage Manager. It positions Tivoli Storage Manager within the Tivoli Storage Management solution, provides an overview of its architecture, the base concepts, the interfaces, and supported environments, and shows Tivoli Storage Manager's interaction with other products of the Tivoli Storage Management product set.

Important: On-line Version Available!

This book-formatted version is valid only for the date of publication.

The most recent version of this material is maintained on the ITSO Internet Web pages at:
`http://www.redbooks.ibm.com/solutions/systemsmanagement/tsm/concepts.html`

1.1 The Tivoli Storage Management solution

In today's connected world, data has become the key asset of companies and one of its most important competitive differentiating factors. Temporary inaccessibility or the complete loss of data has a huge financial impact, and can drive companies out of business. The inability to manage data can have a negative impact on a company's profitability and limit their ability to grow. Storing, protecting, and managing data growth has become one of the major challenges of today's businesses.

Based on this requirement, Tivoli defined its Information Integrity Initiative (in `http://www.tivoli.com/news/press/pressreleases/en/1999/adsm_stormgt.html`): "The Tivoli Storage Management Initiative provides an end-to-end software management solution with proven methodologies to help customers link storage management policies with key business practices, to enable them to use information to DRIVE business, rather than simply support it."

Figure 1 shows the structure of the Tivoli Storage Management Solution and how it fits into the Tivoli Enterprise.

Tivoli Storage Management consists of four major solution components:

- Enterprise protection
- Application protection
- SAN management
- Workgroup protection

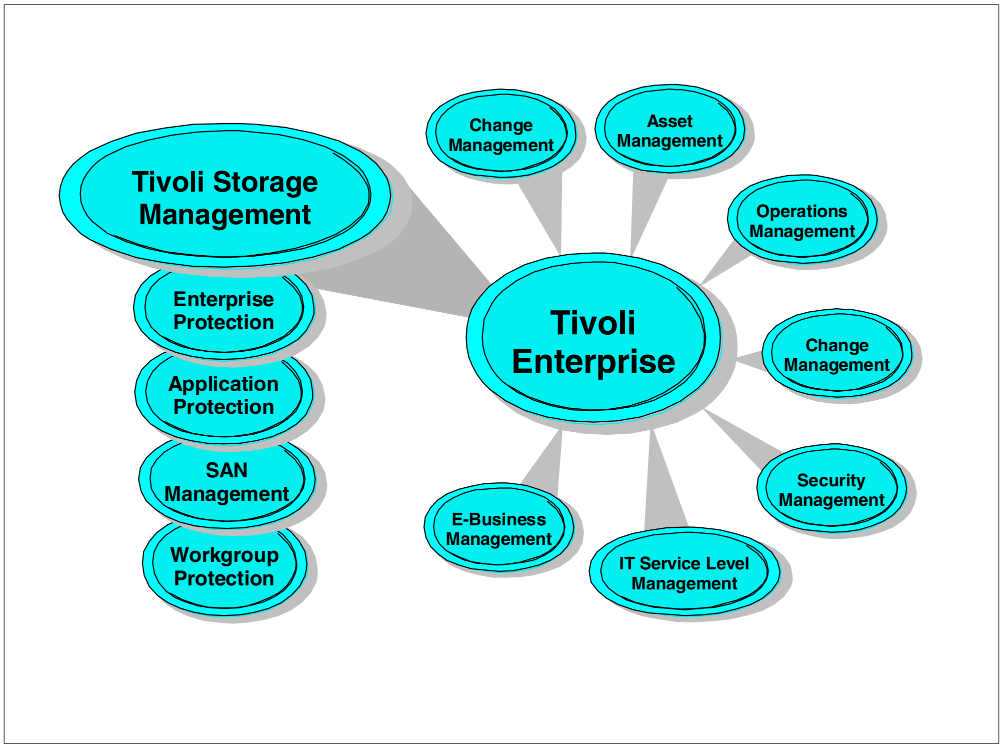


Figure 1. Tivoli Storage Management and Tivoli Enterprise

Enterprise protection implements an enterprise-wide solution for data protection, disaster recovery, space management and record retention. It covers all types of heterogeneous system platforms starting from mobile systems up to large scale enterprise servers and supports all types of storage resources, locally attached as well as network or SAN attached storage. Flexible storage management policies are supporting business needs and powerful automation features by eliminating labor and cost intensive manual storage management tasks.

Strategic business applications are typically complex collections of interdependent components from both commercial and proprietary software and span desktop, distributed and mainframe computing environments. Application protection is concerned with the data availability, performance and recoverability and integrates the application data management into enterprise data protection.

Storage Area Network (SAN) is a new architecture that puts storage on a separate dedicated network to allow businesses of all sizes to provide access to and share data, regardless of operating systems, as a significant step towards helping customers cope with the explosive growth of information in the e-business age. SAN management is concerned with the efficient management of the Fibre Channel based SAN environment. Physical connectivity mapping, switch zoning, performance monitoring, error monitoring, and predictive capacity planning are among the most important features.

Workgroup protection provides a reliable, easy to use, backup, recovery, and disaster recovery solution for stand-alone mobile, desktop and small server systems. It is targeted to small and medium businesses (under 800 nodes) and any enterprise with large numbers of remote, stand-alone servers.

Combined with Tivoli Enterprise, Tivoli Storage Management becomes an integrated management suite that transforms Information Technology into a strategic business resource.

1.2 Tivoli Storage Manager

Tivoli Storage Manager is the core product of the Tivoli Storage Management product set. It provides a solution for distributed data and storage management in an enterprise network environment. It is the next generation of the product originally released by IBM as ADSTAR Distributed Storage Manager (ADSM). Tivoli Storage Manager protects and manages data on more than 30 operating platforms, covering mobile, desktop and server systems over the entire distributed world. It is integrated with hundreds of storage devices as well as LAN, WAN and emerging SAN infrastructures.

The base function provided by Tivoli Storage Manager and its complementary products are:

Data Protection, including:

- **Operational Backup and Restore of Data:** The backup process creates a copy of the data to protect against the operational loss or destruction of file or application data. The customer defines how often to back up (frequency) and how many numbers of copies (versions) to hold.

  The restore process places the backup copy of the data back into a customer-designated system or workstation.

- **Disaster Recovery:** All activities to organize, manage and automate the recovery process from a major loss of IT infrastructure and data across the enterprise. This includes processes to move data offsite into a secure vault location, to rebuild IT infrastructure, and to reload data successfully in an acceptable time frame.

Storage Resource Management, including:

- **Vital Record Retention, Archive and Retrieval:** The archive process creates a copy of a file or a set of files representing an end point of a process for long term storage. Files can remain on the local storage media or can be deleted. The customer controls how long (retention period) an archive copy is to be retained.

  The retrieval process locates the copies within the archival storage and places them back into a customer-designated system or workstation.

- **Hierarchical Space Management:** This process provides the automatic and transparent movement of operational data from the user system disk space to a central storage repository. If the user accesses this data, it is dynamically and transparently restored to the client storage.

The solution is network based, which means that these functions are available to the whole network environment. All the functions can be automated to run in a 24X7 lights-out environment. Administration costs are minimized by centralization of all of the management of Tivoli Storage Manager components.

1.2.1 Tivoli Storage Manager architecture

Tivoli Storage Manager is implemented as a client server software application, consisting of a Tivoli Storage Manager server software component, Tivoli Storage Manager Backup/Archive client and other complementary Tivoli and vendor software products. Figure 2 shows the main components of Tivoli Storage Manager.

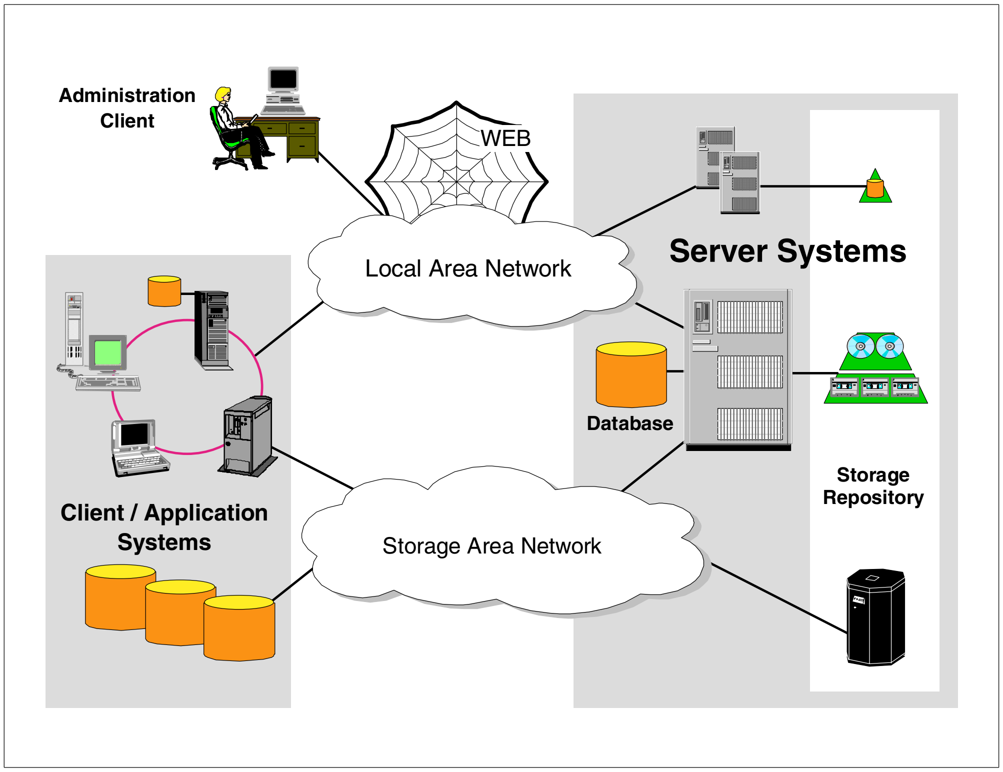


Figure 2. Tivoli Storage Manager architecture

The *Tivoli Storage Manager server software* builds the data management backbone by managing the storage hardware, providing a secure environment, providing automation, reporting and monitoring functions, and implementing the storage management policies and by storing all object inventory information in the *Tivoli Storage Manager database*. The *Tivoli Storage Manager client software* and complementary products implement data management functions like data backup and recovery, archival, hierarchical space management or disaster recovery.

The client software can run on different systems, including laptop computers, PCs, workstations, or server systems. The client and server software can also be installed on the same system for a local backup solution, or to implement LAN-free backup solutions exploiting SAN infrastructure. It is also possible to define server hierarchies or multiple peer-to-peer servers in order to provide a multi-layer storage management solution or an electronic vaulting solution.

1.2.1.1 Tivoli Storage Manager server

One of the principal architectural components of the Tivoli Storage Manager server software is its in-built relational database. The storage manager database was especially designed for the task of managing data, and it implements zero-touch administration. All policy information, logging, authentication and security, media management and object inventory is managed through this database. Most of the fields are externalized through Tivoli Storage Manager high level administration commands, SQL SELECT statements or for reporting purposes, by using an ODBC driver.

For storing the managed data, the Tivoli Storage Manager server uses the *storage repository.* The storage repository is designed from any combination of disk, optical, tape or robotic storage devices, which are locally connected to the server system or which are accessible through a SAN. Exploiting the SAN technology, the server software has features implemented to dynamically share SAN-connected automated tape library systems among multiple Tivoli Storage Manager server systems. The server software provides built-in drivers for more than 300 different device types from every major manufacturer.

Within the storage repository the devices can operate stand-alone or can be linked together to form one or more storage hierarchies. The *storage hierarchy* is not limited in the number of levels and can also span over multiple servers using so-called *virtual volumes*. See 1.2.2.2, “Storage and device concepts” on page 15 for storage management function defined on the storage repository.

1.2.1.2 Tivoli Storage Manager backup/archive client

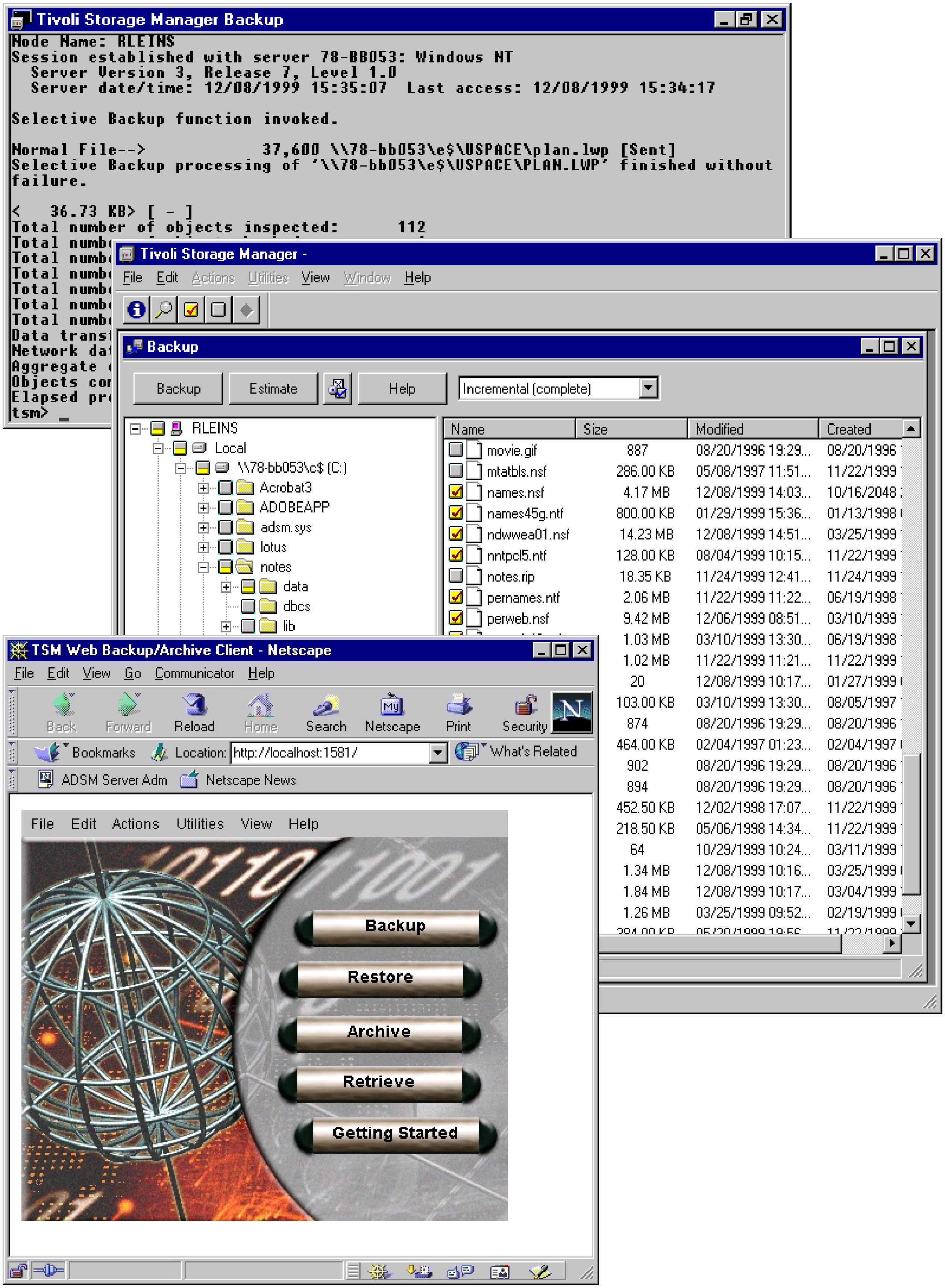
Data management functions are implemented using Tivoli Storage Manager client software and complementary Tivoli and non-Tivoli products, which work together with the Tivoli Storage Manager server backbone product.

The *Tivoli Storage Manager backup/archive client*, included with the server program provides the operational backup and archival function. The software implements the patented *progressive backup methodology* and unique record retention methods as described in 1.2.2.1, “Backup, and archival concepts” on page 13.

All version 3.7 and above backup/archive clients are implemented as multi-session clients, which means that they are able to exploit the multi-threading capabilities of modern operating systems. This enables the running of backup and archive operations in parallel to maximize the throughput to the server system. Full exploitation of multi-threading on the client also requires a 3.7 or higher Tivoli Storage Manager server.

Depending on the client platform the backup/archive client may provide a graphical, command line or Web user interface (see Figure 3). Many platforms provide all three interfaces. The command line interface is useful for experienced users and allows generation of backup or restore scripts for scheduled execution. The graphical interface is designed for ease of use for the end user for ad hoc backups and restores. The web client is especially useful for those clients, such as NetWare, where no native GUI is available, or for performing remote backup/restore operations, for example in a helpdesk environment.

Some UNIX based Version 3.7 clients use a new plug-in architecture to implement an image backup feature for raw device backup. This allows you to backup and recover data that is not stored in file systems or supported database applications. It also provides an additional method to make point-in-time backups of entire file systems as single objects and recover them in conjunction with data backed up by using the progressive backup methodology.



```
Node Name: RLEINS
Session established with server 78-BB053: Windows NT
  Server Version 3, Release 7, Level 1.0
  Server date/time: 12/08/1999 15:35:07  Last access: 12/08/1999 15:34:17

Selective Backup function invoked.

Normal File-->             37,600 \\78-bb053\e$\USPACE\plan.lwp [Sent]
Selective Backup processing of '\\78-bb053\e$\USPACE\PLAN.LWP' finished without
failure.

<   36.73 KB> [ - ]
Total number of objects inspected:      112
Total numb
Total numb
Total numb
Total numb
Total numb
Total numb
Data transf
Network dat
Aggregate d
Objects cor
Elapsed pro
tsm> _
```

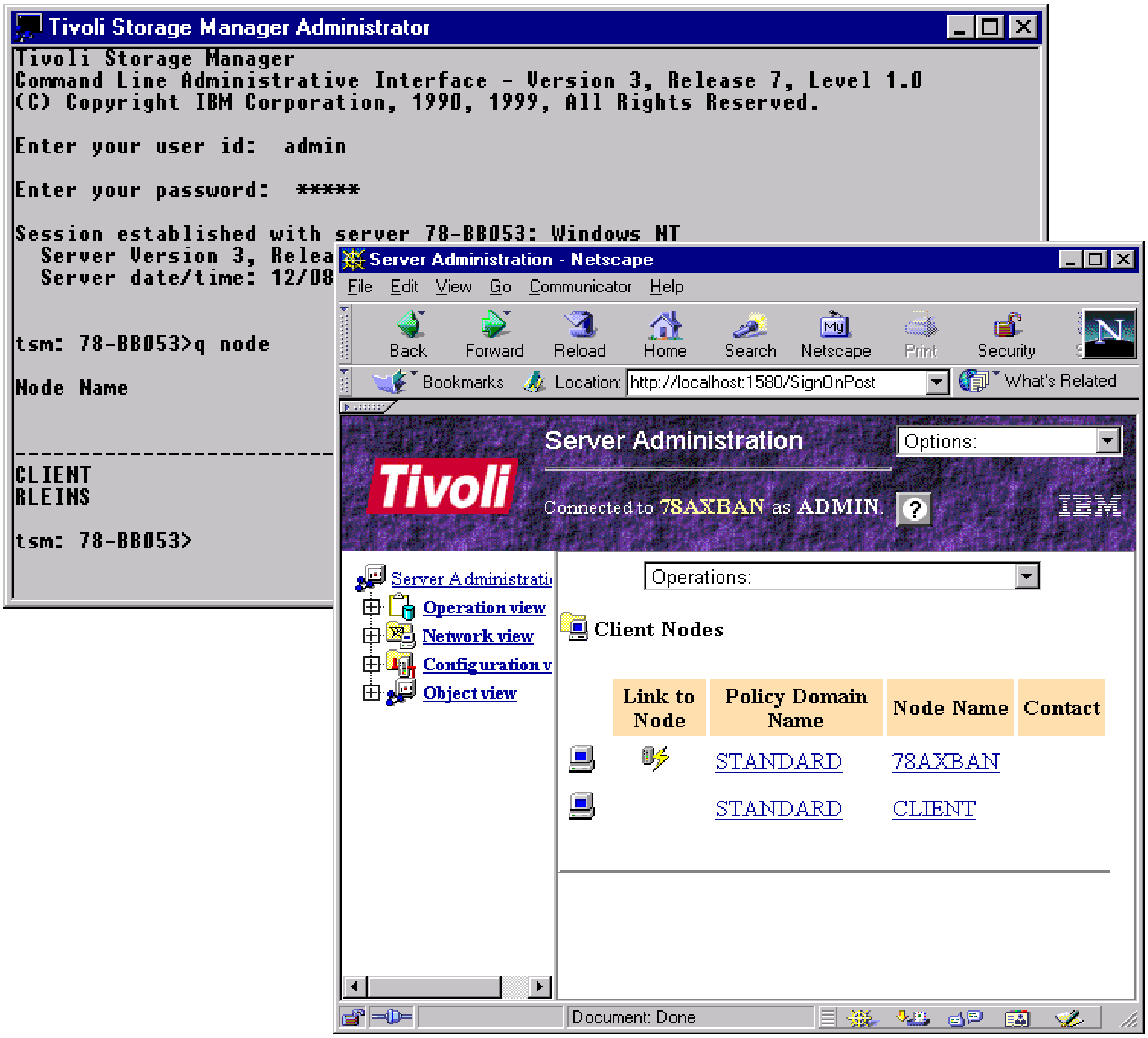
Name	Size	Modified	Created
☐ movie.gif	887	08/20/1996 19:29...	08/20/1996
☐ mtatbls.nsf	286.00 KB	05/08/1997 11:51...	11/22/1999
☑ names.nsf	4.17 MB	12/08/1999 14:03...	10/16/2048
☑ names45g.ntf	800.00 KB	01/29/1999 15:36...	01/13/1998
☑ ndwwea01.nsf	14.23 MB	12/08/1999 14:51...	03/25/1999
☑ nntpcl5.ntf	128.00 KB	08/04/1999 10:15...	11/22/1999
☐ notes.rip	18.35 KB	11/24/1999 12:41...	11/24/1999
☑ pernames.ntf	2.06 MB	11/22/1999 11:22...	06/19/1998
☑ perweb.nsf	9.42 MB	12/06/1999 08:51...	03/10/1999
	1.03 MB	03/10/1999 13:30...	06/19/1998
	1.02 MB	11/22/1999 11:21...	11/22/1999
	20	12/08/1999 10:17...	01/27/1999
	103.00 KB	03/10/1999 13:30...	08/05/1997
	874	08/20/1996 19:29...	08/20/1996
	464.00 KB	02/04/1997 01:23...	02/04/1997
	902	08/20/1996 19:29...	08/20/1996
	894	08/20/1996 19:29...	08/20/1996
	452.50 KB	12/02/1998 17:07...	11/22/1999
	218.50 KB	05/06/1998 14:34...	11/22/1999
	64	10/29/1999 10:24...	03/11/1999
	1.34 MB	12/08/1999 10:16...	03/25/1999
	1.84 MB	12/08/1999 10:17...	03/04/1999
	1.26 MB	03/25/1999 09:52...	02/19/1999

TSM Web Backup/Archive Client - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security
Bookmarks
Location: http://localhost:1581/
What's Related
ADSM Server Adm
Netscape News
File Edit Actions Utilities View Help
Backup
Restore
Archive
Retrieve
Getting Started

Figure 3. Backup/Archive client user interfaces

1.2.1.3 Tivoli Storage Manager administration

For the central administration of one or more Tivoli Storage Manager instances, and the whole data management environment, Tivoli Storage Manager provides command line or Java-based administration interfaces (see Figure 4), also called *administration clients*.



Tivoli Storage Manager Administrator

```
Tivoli Storage Manager
Command Line Administrative Interface - Version 3, Release 7, Level 1.0
(C) Copyright IBM Corporation, 1990, 1999, All Rights Reserved.

Enter your user id:  admin

Enter your password:  *****

Session established with server 78-BB053: Windows NT
  Server Version 3, Relea
  Server date/time: 12/08


tsm: 78-BB053>q node

Node  Name


-------------------------
CLIENT
RLEINS

tsm: 78-BB053>
```

Server Administration - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security
Bookmarks
What's Related
Server Administration
Options:
Tivoli
IBM
Server Administrati
Operation view
Network view
Configuration v
Object view
Operations:

Location: http://localhost:1580/SignOnPost

Connected to 78AXBAN as ADMIN.

Client Nodes

Link to Node	Policy Domain Name	Node Name	Contact
	STANDARD	78AXBAN	
	STANDARD	CLIENT	

Document: Done

Figure 4. Tivoli Storage Manager administration interfaces

Using the unique *enterprise administration* feature it is possible to configure, monitor and manage all server and client instances from one administrative interface, known as the *enterprise console*. It includes:

- Enterprise configuration
- Administrative command routing
- Central event logging functions

The enterprise configuration allows server configurations to be defined centrally by an administrator and then propagated to other servers. This simplifies the configuration and management of multiple Tivoli Storage Manager servers in an enterprise significantly.

Administrative command routing allows administrators to issue commands from one server and route them to other target servers. The commands are executed on the target servers, and the command output is returned and formatted on the server where the command was issued.

In an enterprise environment with multiple Tivoli Storage Manager servers, client and server events can be logged to a central management server through server-to-server communications, thereby enabling centralized event management and automation.

1.2.1.4 Tivoli Storage Manager externalized interfaces

Tivoli Storage Manager provides a data management *application programming interface* (API), which can be used to implement *application clients* to integrate popular business applications, such as databases or groupware applications, into the Tivoli Storage Management solution. The API is also published to allow customers or vendors to implement specialist clients for special data management needs or non-standard computing environments. In general we distinguish between *Tivoli Data Protection for application* software products and the API exploitation through vendor applications.

Tivoli Data Protection for applications are separate program products delivered by Tivoli to connect business applications to the Tivoli Storage Manager data management API. Such applications, for example, Oracle, Lotus Notes and Domino, Microsoft Exchange, and Microsoft SQL server, have their own storage management interfaces which are used to interface to Tivoli Storage Manager. For more information, see 1.4, “Tivoli Data Protection for Applications” on page 21.

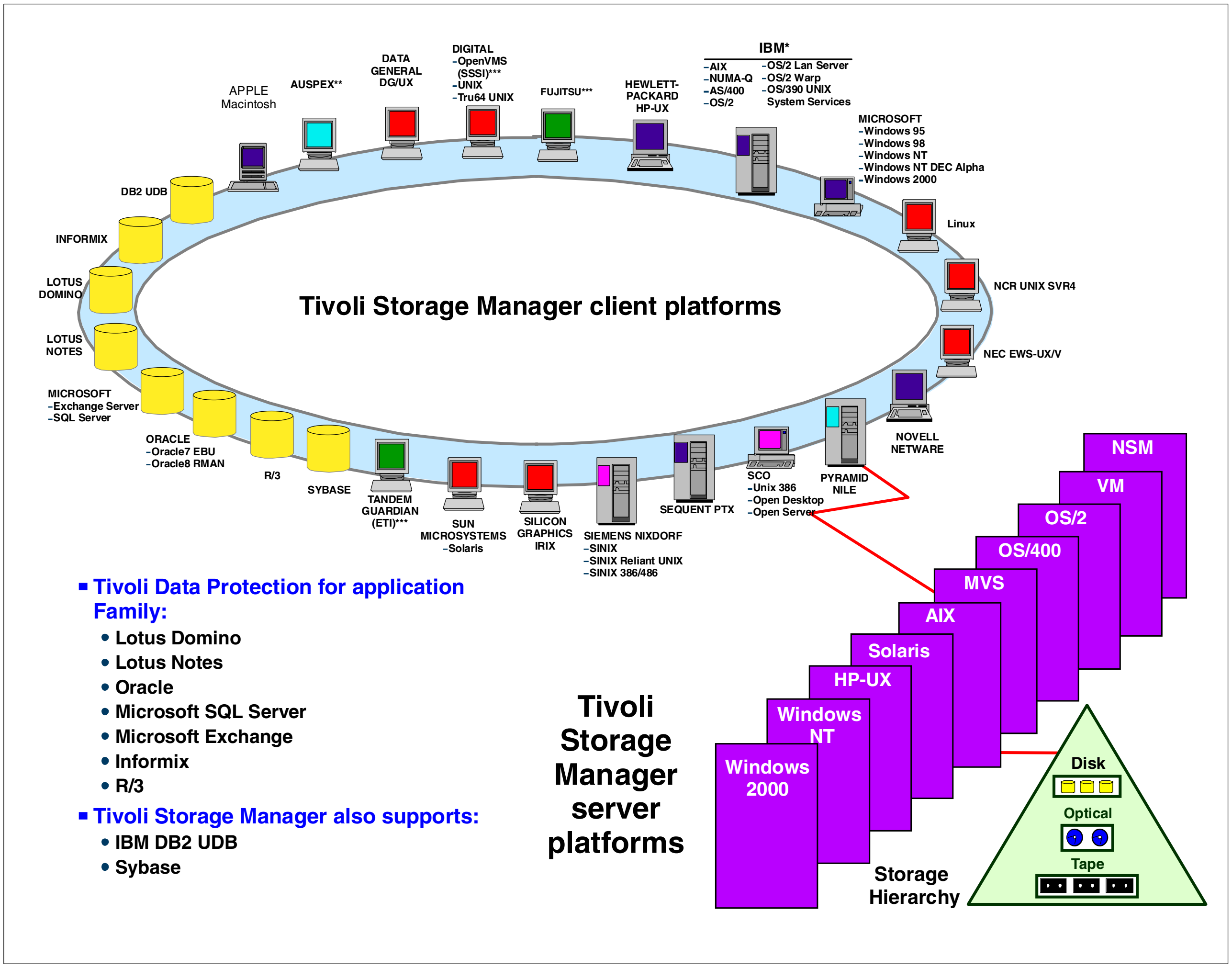
On the other hand, many vendor applications also exploit the Tivoli Storage Manager data management API by integrating it directly into their software product to implement new data management functions, or to provide backup and archival functionality on additional system platforms. Some examples are IBM’s CommonStore for R/3 data archival, IBM’s BRMS/400 to provide an AS/400 backup solution, and SSSI’s Archive Backup Client (ABC) for OpenVMS for data backup and recovery.

In addition to the externalized interfaces to the server database as described in 1.2.1.1, “Tivoli Storage Manager server” on page 6, Tivoli Storage Manager offers multiple interfaces for event logging, reporting and monitoring the data management environment. In general, activities of the Tivoli Storage Manager server and client are logged in the server database and can be sent for reporting and monitoring purposes to external *event receivers* using *event filter* mechanism. Potential event receivers are the Tivoli Enterprise framework, SNMP based systems management software packages, the Windows NT event log, user written applications or others.

To integrate Tivoli Storage Manager storage management with external library management applications Tivoli Storage Manager offers an *external library manager interface*. Using this interface it is possible to integrate the Tivoli Storage Manager server into third-party storage management environments.

1.2.1.5 Tivoli Storage Manager supported environment

Tivoli Storage Manager server and client software is available on many different operating system platforms and can exploit different communication protocols. Figure 7 gives an overview of the supported environment.



- **Tivoli Data Protection for application Family:**
  - Lotus Domino
  - Lotus Notes
  - Oracle
  - Microsoft SQL Server
  - Microsoft Exchange
  - Informix
  - R/3
- **Tivoli Storage Manager also supports:**
  - IBM DB2 UDB
  - Sybase

Figure 5. Tivoli Storage Manager supported environments

The Tivoli Storage Manager server software runs on nine different operating platforms plus the IBM Network Storage Manager (NSM). At this time, the software is available in the most recent Version 3.7 on the platforms, as shown in Table 1, illustrating server platforms, operating system, and Tivoli Storage Manager version.

Table 1. Version 3 servers

Server platforms	Operating system level	Server version
AIX	4.3.1, 4.3.2. 4.3.3	3.7
HP-UX	10.20, 11.0	3.7
MVS/ESA OS/390	4, 5.1 or higher 1 or higher	3.7
Sun Solaris	2.6, 7	3.7

Server platforms	Operating system level	Server version
Windows NT	Workstation 4.0 (SP3/4) Server 4.0 (SP3/4)	3.7
Windows2000		3.7.3
OS/400	V4R3 or higher	3.1.2
VM/ESA	2.2,2.3,2.4	3.1
OS/2	WARP 4.0 or higher WARP Server 4.0	2.1

The following tables provide an overview of all available Version 3.7 clients at the time of publishing this book. Other operating system clients (available at pre-3.7 levels) may be updated in the future. Check the product information on the Tivoli Storage Manager home page for the latest complete client availability information:

`http://www.tivoli.com/products/index/storage_mgr/`

There are several variations of UNIX clients. Table 2 details the UNIX clients and the operating system levels that are supported.

Table 2. Version 3.7 UNIX clients

Client Platforms	Operating System
AIX	4.3.1, 4.3.2, 4.3.3
HP-UX	11.0
Sun Solaris	2.6, 7
Linux	SUSE 6.3, RedHat 6.1, Turbo Linux 6.0, Calder Linux 2.3
OS/390 UNIX	Version 2 Release 6 or Version 2 Release 7
SCO UnixWare	7.01
Tru64	4.0d or higher
SGI IRIX	6.5.x
IBM NUMA-Q (formerly Sequent)	DYNIX/ptx 4.5.1 or higher

There are also different PC clients available. Table 3 details the PC systems and the operating systems that are supported as clients.

Table 3. Version 3.7 PC clients

PC Clients Platforms	Operating Systems
Novell NetWare	3.12, 3.20, 4.11, 4.20, 5.0
Microsoft Windows (Intel)	NT 4.0, Win 95, Win 98, Win 2000
Microsoft Windows (Alpha)	NT 4.0
Apple Macintosh for PowerPC	System 7.5.5, MacOS 8.x, MacOS 9.x
IBM OS/2	Warp 4.0 or later

1.2.2 Base concepts

This section gives a high level introduction to the base data and storage management paradigms used by Tivoli Storage Manager to implement its functionality. We will cover data protection or backup, record retention or archival, storage management, policy, and security.

1.2.2.1 Backup, and archival concepts

Backup, in Tivoli Storage Manager terms, means creating an additional copy of a data object to be used for recovery. A data object can be a file or a user defined data object like a database table. The backup version of this data object is stored separately in the Tivoli Storage Manager server storage repository. Potentially, you can make several backup versions of the data, each version at a different point-in-time. These versions are closely tied together and related to the original object as a group of backups.

If the original data object is corrupted or lost on the client system, r*estore* is the process of sending a backup version of the data from the server back to the client. The most current version of the data is normally restored, but you can choose to restore from any of the existing backup versions. The number of backup versions is controlled by server definitions. Old backup versions may be automatically deleted as new versions are created. You may also delete them after a certain period of time.

For file level based backup the main difference from many other backup applications is that Tivoli Storage Manager uses the *progressive backup methodology*. As shown in Figure 6, after the first necessarily full backup, Tivoli Storage Manager then operates with incremental backups only. In consequence, only those files that have changed since the last backup will be backed up.

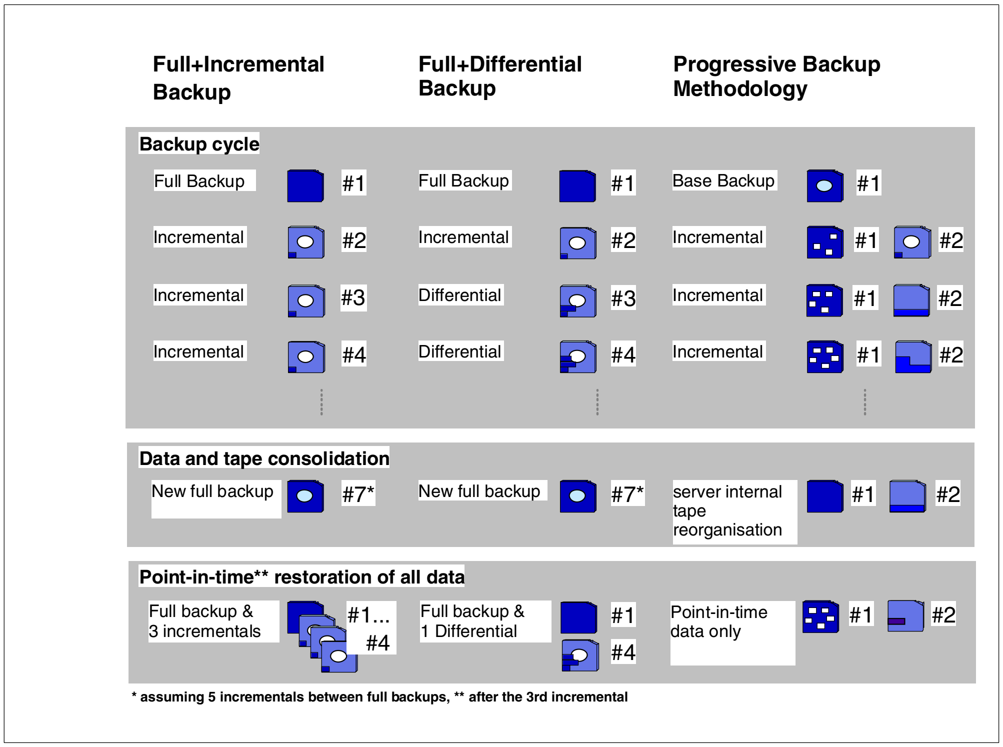

Full+Incremental Backup
Full+Differential Backup
Progressive Backup Methodology
Backup cycle
Full Backup #1
Incremental #2
Incremental #3
Incremental #4
Full Backup #1
Incremental #2
Differential #3
Differential #4
Base Backup #1
Incremental #1 #2
Incremental #1 #2
Incremental #1 #2
Data and tape consolidation
New full backup #7*
New full backup #7*
server internal tape reorganisation #1 #2
Point-in-time** restoration of all data
Full backup & 3 incrementals #1... #4
Full backup & 1 Differential #1 #4
Point-in-time data only #1 #2
* assuming 5 incrementals between full backups, ** after the 3rd incremental

Figure 6. Progressive Backup Methodology vs. other backup schemes

The reorganization of the physical storage media to store each client's data physically together on a small number of media—in order to provide faster access in the case of a complete system recovery—is done transparently to the client, and is completely automated on the server using data meta information stored in the server database.

Tivoli Storage Manager's file level progressive backup methodology, in comparison with other methods like Full+Incremental or Full+Differential backup schemes, prevents unnecessary backups of unchanged data to reduce and consolidate the recovery tape-set. It offers also a more efficient use of storage resources by not storing redundant data, and a faster recovery by not restoring multiple versions of the same file.

At any point in time Tivoli Storage Manager allows the creation of a complete set of client files (*backup set*) on the server system using the most recent backup versions stored in the server storage repository. These backup sets can be used to retain a snapshot of all client files for a longer period of time (*Instant Archive*) or for LAN-free recovery of a client system by copying this backup set onto portable media and restoring them locally (*Rapid Recovery*).

File *Archive* with Tivoli Storage Manager means creating a copy of a file as a separate object in the storage repository to be retained for a specific period of time. Typically you would use this function to create an additional copy of data to be saved for historical purposes. Vital records (data that must be kept for legal or other business reasons) are likely candidates for the archive process. You can specify to delete the original copy of the data on the source system once the archive copy is created on the server. Therefore, you can use *archive* to make additional space available on the Tivoli Storage Manager client system. However, *archive* should not be thought of as a complete space management function, because transparent automatic recall is not available.

You can access archived data by using *retrieve* to return it to the Tivoli Storage Manager client, if the data is needed at some future time. To locate the archived data within the storage repository, Tivoli Storage Manager allows you to add a description to the data and to form *archive packages* of related files. You can then use this description to search the server database for matching packages, to determine which data to retrieve.

Therefore, the difference between *backup* and *archive* is that backup creates and controls multiple backup versions that are directly attached to the original file; whereas archive creates an additional file that is normally kept for a specific period of time, as in the case of vital records.

1.2.2.2 Storage and device concepts

All Tivoli Storage Manager-managed client data are stored in the Tivoli Storage Manager storage repository, which can consist of different storage devices, such as disk, tape, or optical devices, and controlled by the Tivoli Storage Manager server. To do this, Tivoli Storage Manager uses its own model of storage to view, classify, and control these storage devices, and to implement its storage management functionality (see Figure 7).

The main difference between the storage management approach of Tivoli Storage Manager and other commonly used systems is that Tivoli Storage Manager storage management concentrates on managing *data objects* instead of managing and controlling backup tapes. Data objects can be files, directories or raw logical volumes that are backed up from the client systems; they can be objects like tables or records from database applications, or simply a block of data that a client system wants to store on the server storage.

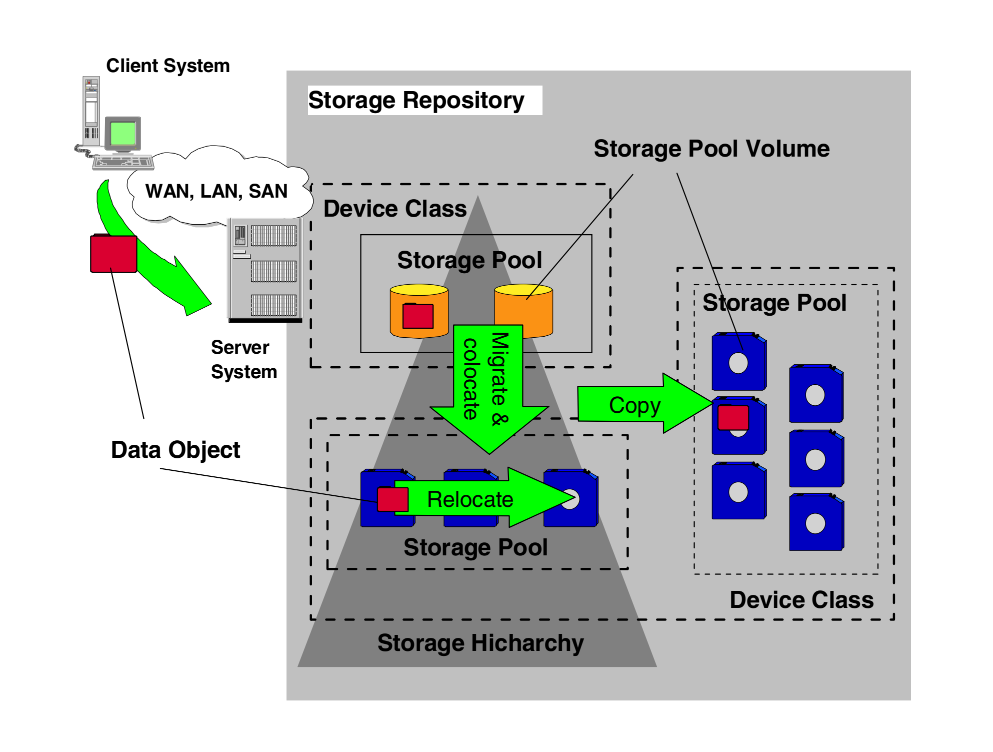


Figure 7. Tivoli Storage Manager storage management concept

To store these data objects on storage devices and to implement storage management functions, Tivoli Storage Manager has defined some logical entities to classify the available storage resources. Most important is the logical entity called a *storage pool.* A storage pool describes a storage resource for one single type of media; for example, a disk partition or a set of tape cartridges. Storage pools are the place where data objects are stored.

A storage pool is built up from one or more *storage pool volumes*. For example, in the case of a tape storage pool, this would be a single physical tape cartridge. To describe how Tivoli Storage Manager can access those physical volumes to place the data objects on them, Tivoli Storage Manager has another logical entity called a *device class*. A device class is connected to a storage pool and specifies how volumes of this storage pool can be accessed.

Tivoli Storage Manager organizes storage pools in one or more hierarchical structures. This *storage hierarchy* can span over multiple server instances and is used to implement management functions to *migrate* data objects automatically—completely transparent to the client—from one storage hierarchy level to another; or in other words, from one storage device to another. This function may be used, for example, to cache backup data (for performance reasons) onto a Tivoli Storage Manager server disk space before moving the data to tape cartridges. The actual location of all data objects is automatically tracked within the server database.

Tivoli Storage Manager has implemented additional storage management functions for moving data objects from one storage volume to another. As discussed in the previous section, Tivoli Storage Manager uses the progressive backup methodology to backup files to the Tivoli Storage Manager storage repository. The reorganization of the data and storage media for fast recovery happens completely within the server. For this purpose, Tivoli Storage Manager has implemented functions to *relocate* data objects from one volume to another and to *collocate* data objects that belong together, either at the client system level or at the data group level.

Another important storage management function implemented within the Tivoli Storage Manager server is the ability to *copy* data objects asynchronously and to store them in different storage pools or on different storage devices, either locally at the same server system or remotely on another server system. It is especially important for disaster recovery reasons to have—in the event of losing any storage media or the whole storage repository—a second copy of data available somewhere in a secure place. This function is fully transparent to the client, and can be performed automatically within the Tivoli Storage Manager server.

1.2.2.3 Policy concepts

A data storage management environment consists of three basic types of resources: client systems, rules, and data. The client systems contain the data to be managed, and the rules specify how the management must occur; for example, in the case of backup, how many versions should be kept, where they should be stored, and so on.

Tivoli Storage Manager policies define the relationships between these three resources. Figure 8 illustrates this policy relationship. Depending on your actual needs for managing your enterprise data, these policies can be very simple or very complex.

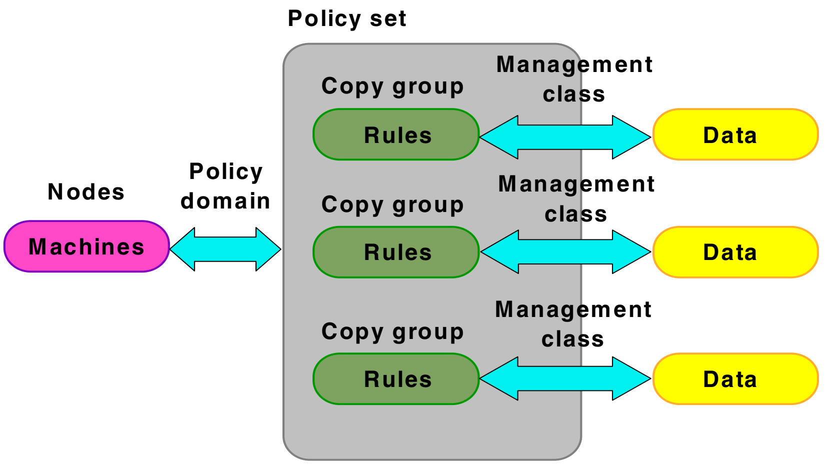


Figure 8. Policy relationships and resources

Tivoli Storage Manager has certain logical entities that group and organize the storage resources and define relationships between them. Client systems, or nodes in Tivoli Storage Manager terminology, are grouped together with other nodes with common storage management requirements, into a *policy domain*.

The policy domain links the nodes to a *policy set*, a collection of storage management rules for different storage management activities. A policy set consists of one or more *management classes*. A management class contains the rule descriptions called *copy groups*, and links these to the data objects to be managed. A copy group is the place where all the storage management parameters, such as number of stored copies, retention period, storage media, and so on, are defined. When the data is linked to particular rules, it is said to be bound to the management class that contains those rules.

Another way to look at the components that make up a policy is to consider them in the hierarchical fashion in which they are defined. That is to say, consider the policy domain containing the policy set, the policy set containing the management classes, and the management classes containing the copy groups and the storage management parameters.

1.2.2.4 Security concepts

Because the storage repository of Tivoli Storage Manager is the place where all the data of an enterprise are stored and managed, security is a very vital aspect for Tivoli Storage Manager. To ensure that data can only be accessed from the owning client or an authorized party, Tivoli Storage Manager implements, for authentication purposes, a *mutual suspicion algorithm*, which is similar to the methods used by Kerberos authentication.

Whenever a client (backup/archive or administrative) wants to communicate with the server, an authentication has to take place. This authentication contains *both-sides* verification, which means that the client has to authenticate itself to the server, and the server has to authenticate itself to the client.

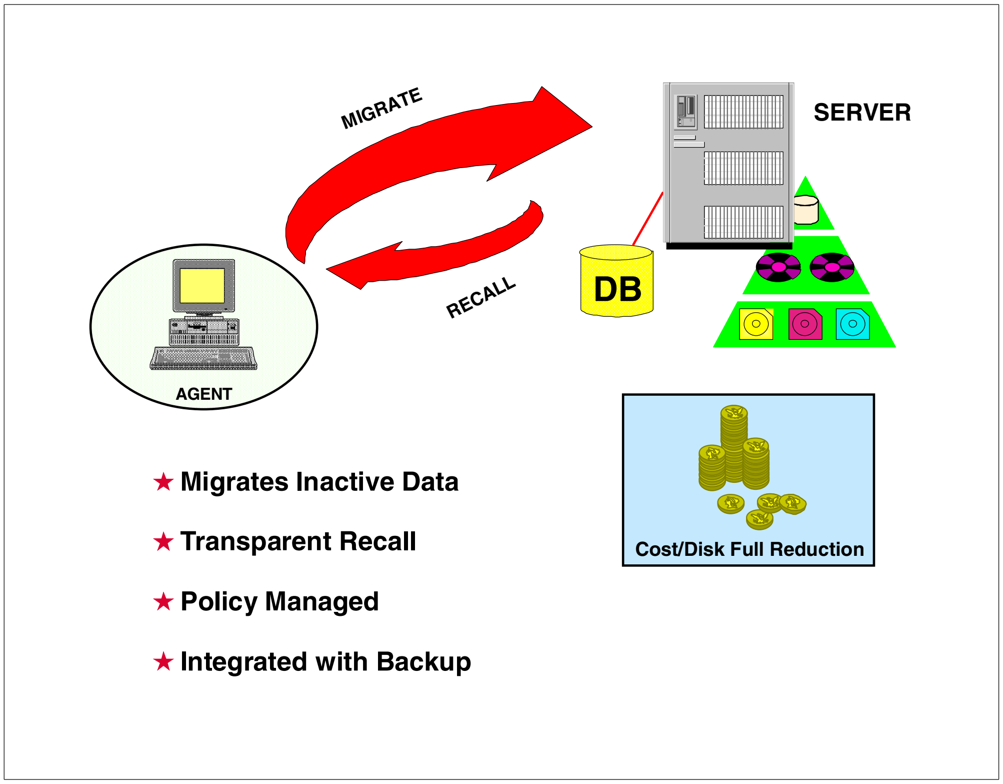
To do this, all clients have a password, which is stored at the server side as well as at the client side. In the authentication dialog these passwords are used to encrypt the communication. The passwords are not sent over the network, to prevent hackers from intercepting them. A communication session will be established only if both sides are able to decrypt the dialog. If the communication has ended, or if a timeout period without activity is passed, the session will be automatically terminated and a new authentication will be necessary.

1.3 Tivoli Storage Manager complementary products

Tivoli Storage Manager complementary products use the Tivoli Storage Manager server software package as a backbone product to implement additional data and storage management functions. The following section introduces Tivoli Space Manager for hierarchical space management, Tivoli Disaster Recovery Manager as an enterprise-wide solution for disaster recovery and Tivoli Decision Support for Storage Management Analysis for a comprehensive reporting and monitoring solution to plan the growth and collect vital management information for an efficient enterprise data management deployment.

1.3.1 Tivoli Space Manager

Tivoli Space Manager uses the framework services of Tivoli Storage Manager in combination with the industry standard *Data Management Application Programming Interface (DMAPI)* to deliver a fully integrated solution for open systems *Hierarchical Space Management (HSM)*. Tivoli Space Manager provides an HSM client, which interfaces with DMAPI and implements the functionality outlined in Figure 9.



★ Migrates Inactive Data

★ Transparent Recall

★ Policy Managed

★ Integrated with Backup

Figure 9. Hierarchical storage management

Tivoli Space Manager maximizes usage of existing storage resources by transparently migrating data from client hard drives to the Tivoli Storage Manager storage repository based on size and age criteria. When the migrated data is accessed, Tivoli Space Manager transparently recalls it back onto the local disk.

The migration of files and the management of migrated files is controlled by policies. However user controlled migration and recall is also possible.

Tivoli Space Manager’s HSM function is fully integrated with Tivoli Storage Manager operational backup. It is possible to specify not to migrate a file until it has a backup version in the server storage repository. If a file is migrated and then a backup is done the next day, Tivoli Storage Manager copies the file from the Space Migration pool to the backup pool, instead of requiring a recall to the client system to back it up again, which would cause multiple transfers of the same file across the network.

1.3.2 Tivoli Disaster Recovery Manager

Tivoli Disaster Recovery Manager coordinates and automates the process of recovering from a disaster. It integrates with Tivoli Storage Manager and the rest of the Tivoli data management portfolio to provide for offsite media management, automated restore and managed recovery. It complements the already implemented robust protection features of Tivoli Storage Manager and automates many already facilitated protection functions.

Tivoli Disaster Recovery Manager automatically captures information required to recover the Tivoli Storage Manager server after a disaster. It assists in preparing a plan that allows recovery in the most expedient manner. This *disaster recovery plan* contains information, scripts, and procedures needed to automate server restoration, and helps ensure quick recovery of your data after a disaster.

Tivoli Disaster Recovery Manager also manages and tracks the movement of off-site media to reduce the time required to recover in the event of a disaster. It is able to track media that are stored on-site, in-transit, or off-site in a vault, no matter whether it is a manual or electronic vault, so your data can be easily located if disaster strikes.

Client recovery information can also be captured by Tivoli Disaster Recovery Manager to assist with identifying what clients need to be recovered, in what order, and what is required to recover them.

1.3.3 Tivoli Decision Support for Storage Management Analysis

Tivoli Decision Support for Storage Management Analysis uses the framework services of Tivoli Decision Support to deliver important decision-making information about your enterprise data management deployment.

Tivoli Decision Support is a stand-alone product that provides a ready-to-use view into the wealth of data gathered by Tivoli enterprise products. The product consolidates this data from Tivoli products and transforms it into accessible IT business-relevant information. This information, presented in a variety of graphical formats can be viewed interactively (slice, dice, drill down, drill through) and posted on a URL. Tivoli Decision Support provides insight and the ability to better answer IT business-relevant questions. Tivoli Decision Support is available on Windows NT.

The Tivoli Decision Support Discovery Guides are a set of best practices guides provided for various applications. To use these guides Tivoli Decision Support has to be installed and available. The Tivoli Decision Support for Storage Management Analysis is the guide used to produce the following analyses:

- Storage Event Analysis
- Storage Performance Analysis
- Storage Capacity Analysis

The architecture and the information flow is shown in Figure 10.

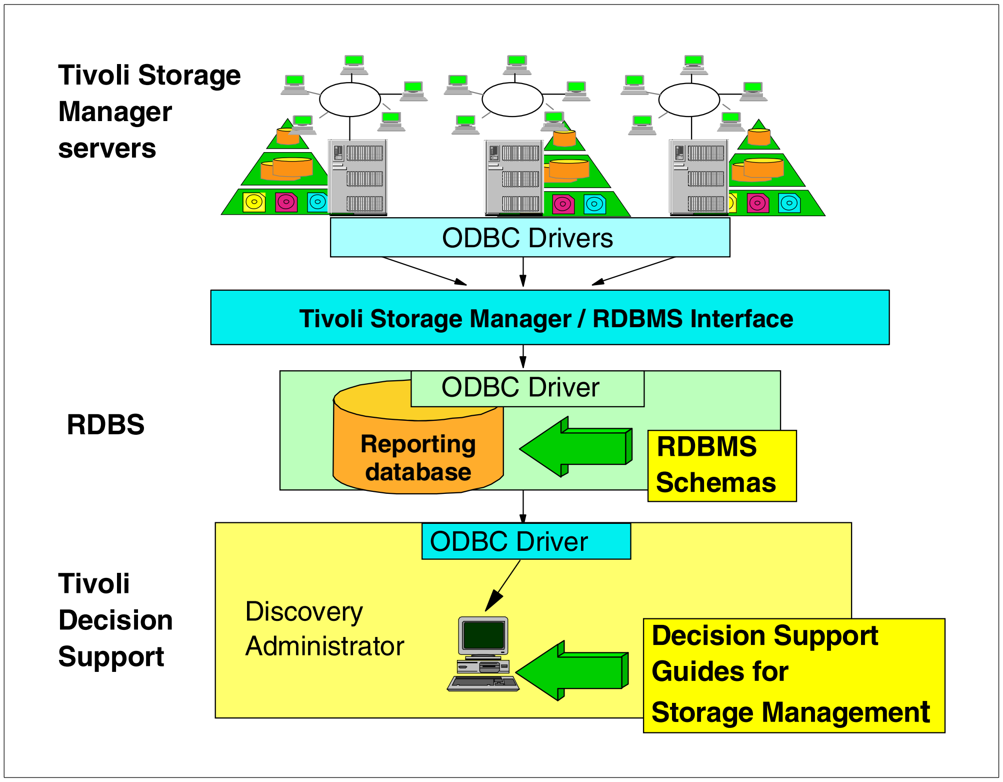

Tivoli Storage Manager servers
ODBC Drivers
Tivoli Storage Manager / RDBMS Interface
RDBS
ODBC Driver
Reporting database
RDBMS Schemas
Tivoli Decision Support
ODBC Driver
Discovery Administrator
Decision Support Guides for Storage Management

Figure 10. Tivoli Decision Support for Storage Management Analysis

The information used by the guide is obtained directly from the Tivoli Storage Manager server with the use of the ODBC interface. The information is then transferred to a relational database, another requirement for Tivoli Decision Support for Storage Management Analysis. The databases supported for the information feed are DB2, MS SQL and Oracle. The database can reside on the same system as Tivoli Storage Manager or Tivoli Decision Support or on a separate system. The database is used to for queries to generate the Tivoli Decision Support reports.

1.4 Tivoli Data Protection for Applications

Tivoli Data Protection for Applications is a group of solutions integrated to Tivoli Storage Manager, which protect data used by business applications. They are interface programs between a storage management API provided by the vendor application, and the Tivoli Storage Manager data management API. Typical applications providing such interfaces are databases and groupware applications, such as Lotus Notes or Microsoft Exchange.

Figure 11 shows Tivoli Data Protection for Lotus Domino as a typical example of the architecture and the data flow of a Tivoli Data Protection for application solution.

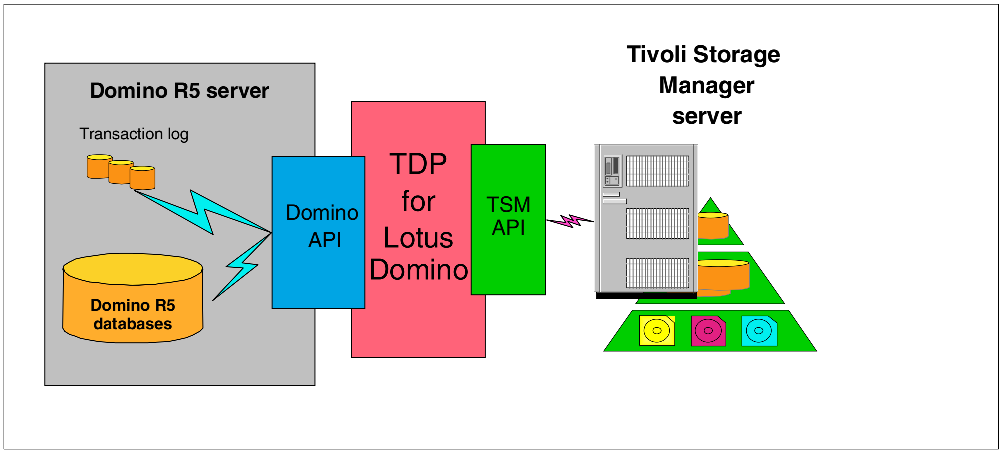


Figure 11. Tivoli Data Protection for Lotus Domino

The function of the Tivoli Data Protection for Applications solutions is to receive application backup and restore requests and to translate them into Tivoli Storage Manager backups and restores. The activity is always initiated from the application. This means that backups or restores can also be done while the application is on line. The TDP application client therefore implements the intelligence and function of the backup interface provided by the application itself.

Table 4 shows the available Tivoli\ Data Protection for application solutions and the platforms, operating system level and application level they cover at the time of publishing this book.

Table 4. Tivoli Data Protection for application

TDP solution	Application level	Operating system	Operating system level
TDP for Lotus Notes	4.5.x, 4.6.0, 4.6.1, 4.6.3 - 4.6.6	AIX NT	4.2.x, 4.3.x 4.0 SP4
TDP for Lotus Domino	5.0.1	AIX NT Windows 2000 Solaris	4.3.1, 4.3.2 4.0 SP4 2.6 or 7
TDP for Lotus Domino, S/390 Edition	5.0.1	OS/390	2.6, 2.7 with UNIX System services
TDP for MS Exchange	5.0, 5.5	NT Windows 2000	4.0 SP4
TDP for MS SQL Server	6.5, 7.1	NT Windows 2000	4.0 SP4
TDP for Informix	IDS 7, UDO 9	AIX Sun Solaris	4.3.2, 4.3.3 2.6, 7
TDP for Oracle	7.3.4, 8.0.X, 8.1.5, 8.1.6	AIX NT Sun Solaris HP-UX Windows 2000	4.3.1, 4.3.2, 4.3.3 4.0 2.6, 7 (64bit) 11.0

TDP solution	Application level	Operating system	Operating system level
TDP for /R3	3.0C to F, 3.1 up to Z, 4.0 up to 4.6C	AIX HP-UX Sun Solaris Tru64 UNIX NT (Intel) LINUX	4.3 11.0 7 4.0 4.0 RedHat 6.1 EE

For the latest information on Tivoli Data Protection solutions, check the following Tivoli Storage Management Web page:

`http://www.tivoli.com/products/solutions/storage/`

1.5 Tivoli Data Protection for Workgroups

Tivoli Data Protection for Workgroups provides a complete disaster and bare metal recovery solution for Windows NT servers.

It uses Stac's unique *Object Replication Technology* to ensure backup of all disk contents, such as partition information, boot records, FAT and NTFS partitions, while allowing full read and write access to the system during backup. It can backup entire Windows NT servers or volumes at near maximum device speed to local tape or disk drives.

Tivoli Data Protection for Workgroups also provides the function to create a set of bootable diskettes that may be used in the event of hard drive failure or other disaster.

The restore operation can be performed using the Windows File Explorer for single file restores by simply drag and drop files from the mounted backup medium to their original location or for more complex restores by utilizing a Tivoli Data Protection for Workgroups administrative tool. In the event of a complete disk failure the set of bootable diskettes can be used to perform a bare-metal restore of the entire system.

The product can be either used stand-alone or integrated in a Tivoli Storage Manager environment for the following functions:

- Message logging to a Tivoli Storage Manager server, using the Tivoli Storage Manager API client.
- Automate operations with the scheduler component provided by the Tivoli Storage Manager backup-archive client.

Part 2. Standard base function

Part 2 covers the base Tivoli Storage Manager functions that all implementations must use.

It excludes optional features that are licensed separately. These are covered in Part 3.

Pointer to "how-to" information:

An ITSO "how-to" guide is available as a companion to this conceptual information. It provides material describing how to implement and operate a basic setup of Tivoli Storage Manager Version 3.7. It is available as a hardcopy redbook, Getting Started with Tivoli Storage Manager: Implementation Guide, SG24-5416 or in softcopy at `http://ibm.com/redbooks`.

Chapter 2. Backup-Archive client

This chapter covers the main client concepts to perform backup and restore operations. For further details on implementation, see *Getting Started with Tivoli Storage Manager: Implementation Guide*, SG24-5416.

The Backup-Archive client is the software program that helps you protect information on your workstation. Tivoli Storage Manager lets you submit and receive information, to and from a Tivoli Storage Manager server, by controlling the transmission back and forth. You use the Tivoli Storage Manager Backup-Archive client to maintain backup versions of a machine's files. Then you can recover older file versions in the event those files are lost or damaged.

2.1 What is a client?

Tivoli Storage Manager is a client-server program. The client product must be installed in the machine you want to back up. The client portion is responsible for sending and receiving data to and from the Tivoli Storage Manager server.

The Backup-Archive client has two distinct features:

- The backup feature allows users to back up (and manage) a number of versions of their data onto the Tivoli Storage Manager server (or servers) and to restore from these, if the original files are lost or damaged. Some examples of loss or damage are hardware failure, theft of computer system, or virus attack.
- The archive feature allows users to keep a copy of their data for long term storage and to retrieve the data if necessary. Examples of this need may be to meet legal requirements, to return to a previous working copy if the software development of a program is unsuccessful, or to archive files that are not currently needed on a workstation.

Figure 12 shows a client machine (HARRY) and a Tivoli Storage Manager server (SRV_TSM1). The client machine interacts with the server to send and receive data.

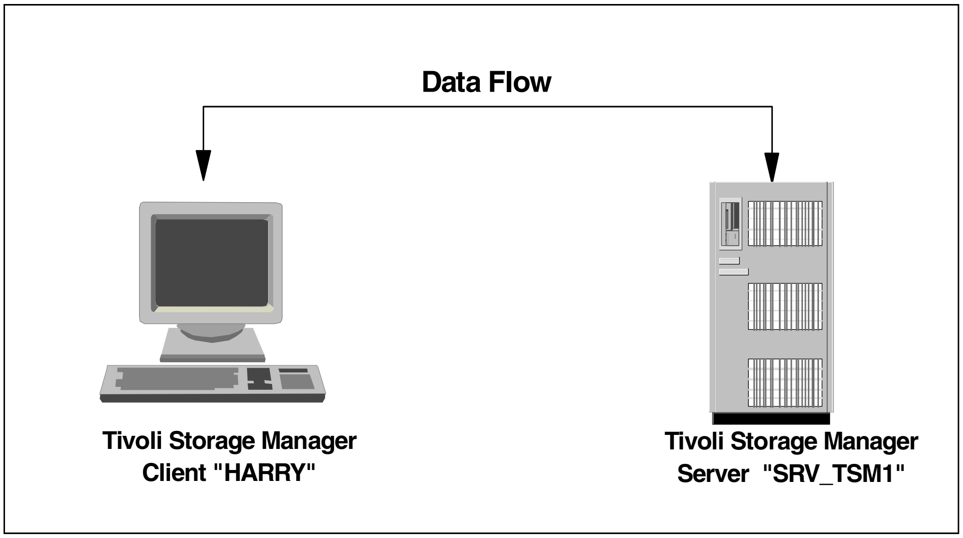


Figure 12. How a client computer interacts with Tivoli Storage Manager

2.2 Client components

Each client has two major components which help you protect your important data:

1. Software components: These are the software programs and customization files that you must have in place to use Tivoli Storage Manager. The most important of these are the client interfaces. Each interface is designed so that you can perform all client operations from the one that best suits your needs.

   Client interfaces are flexible. You may back up files on a UNIX system and then restore them onto another Tivoli Storage Manager UNIX platform. You may back up file allocation table (FAT) files from DOS, Windows, OS/2, Windows NT, and then restore them to another workstation running one of the above-mentioned operating systems. In general, cross-client recover can be done among clients with compatible file system formats. Adjustments may need to be made in file name length (like OS/2 long file name restore to DOS) and file security. For successful interaction with the server, you must configure some basic parameters in a client options file.

2. Operation components: When you use the Tivoli Storage Manager interface to back up or archive a file, it sends a copy of the file and its associated attributes to the Tivoli Storage Manager server. The Tivoli Storage Manager client can perform two types of operations: backups and archives. Although these have different purposes, you can think of them as the alternatives that you have to better control how data must be saved.

   All client processing is controlled and secure. A client can restore or retrieve their own files, or the files that they have been authorized to restore by the owner. Whenever a client communicates with the server, it starts a new session. The Tivoli Storage Manager server tracks sessions and each client activity.

The Backup-Archive client can be started manually by the user, by an Administrator, or even automatically by the operating system.

The Backup-Archive client provides support for errors such as communications errors, unreadable client files during backups, or volumes that are unavailable on the Tivoli Storage Manager server during restore.

2.2.1 Interfaces

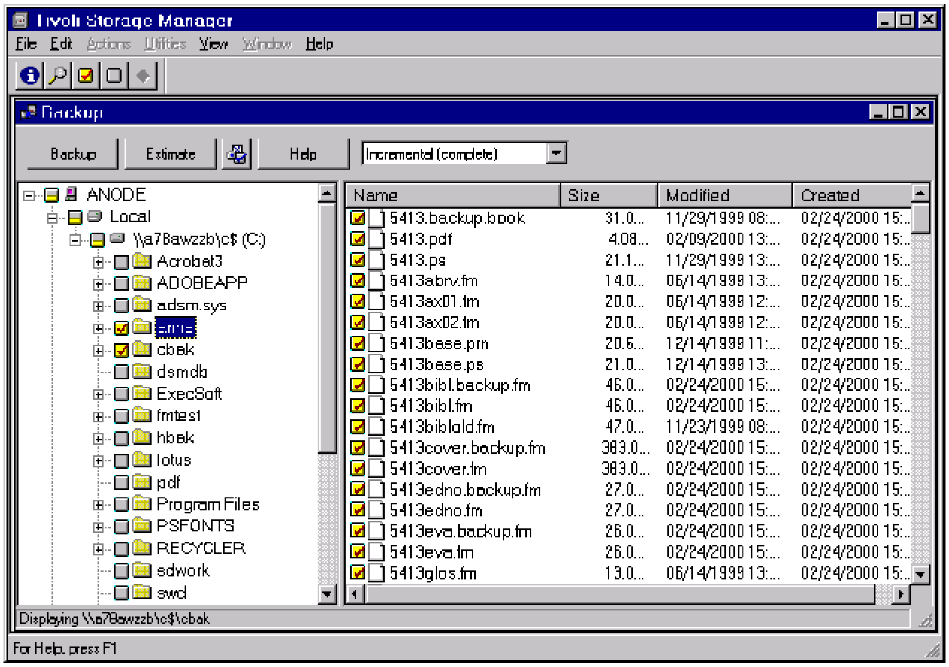
There are three ways you can interact with the Tivoli Storage Manager server to run a backup or archive operation:

- Graphical User Interface (GUI)
- Command Line Interface (CLI)
- Web Client Interface (Web Client)

There are minor differences between the Backup-Archive client code among the platforms. For example, in Windows NT you have specific options to handle the Windows NT Registry information, which is not found in any other platform. You use the native file formats to specify filespaces for backup (for example /usr in UNIX or \\hostname\d$ for Windows). Apart from these OS specific considerations, all commands are the same on each client platform.

2.2.1.1 GUI

The GUI is a much more user-friendly interface, with unique features for the end-user. You may use it for backup, restore, archive, and retrieve operations, and it is intuitive for you to understand which files and directories you are backing up. The GUI is slightly different between Windows NT and UNIX machines. In spite of the appearance, UNIX and Windows Backup-Archive clients have the same functionality. Figure 13 shows a Windows NT Backup-Archive client screen.



Incremental (complete)

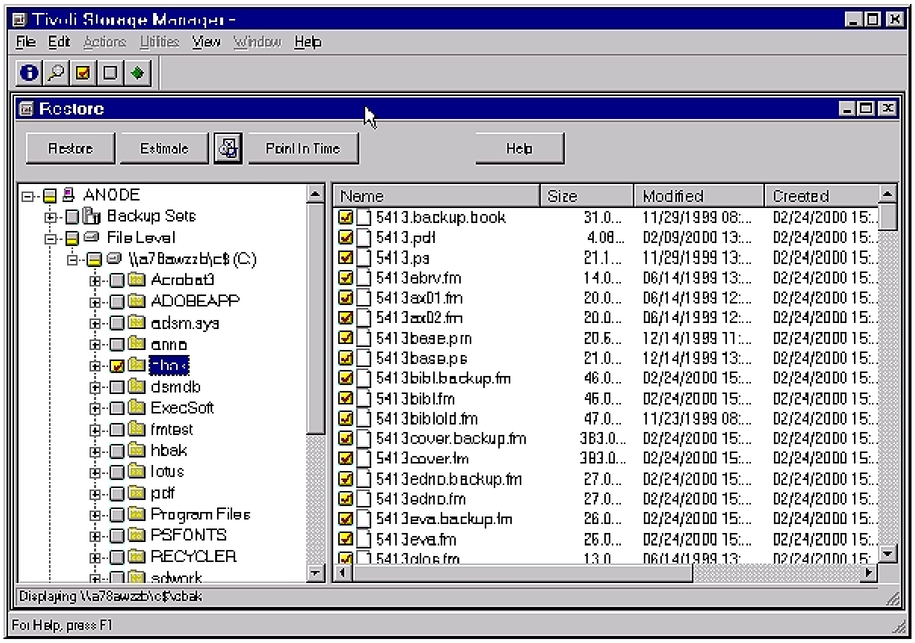
	Name	Size	Modified	Created
☑	5413.backup.book	31.0...	11/29/1999 08:...	02/24/2000 15:...
☑	5413.pdf	4.08...	02/09/2000 13:...	02/24/2000 15:...
☑	5413.ps	21.1...	11/29/1999 13:...	02/24/2000 15:...
☑	5413abrv.fm	14.0...	06/14/1999 13:...	02/24/2000 15:...
☑	5413ax01.fm	20.0...	06/14/1999 12:...	02/24/2000 15:...
☑	5413ax02.fm	20.0...	06/14/1999 12:...	02/24/2000 15:...
☑	5413base.pm	20.6...	12/14/1999 11:...	02/24/2000 15:...
☑	5413base.ps	21.0...	12/14/1999 13:...	02/24/2000 15:...
☑	5413bibl.backup.fm	46.0...	02/24/2000 15:...	02/24/2000 15:...
☑	5413bibl.fm	46.0...	02/24/2000 15:...	02/24/2000 15:...
☑	5413biblold.fm	47.0...	11/23/1999 08:...	02/24/2000 15:...
☑	5413cover.backup.fm	383.0...	02/24/2000 15:...	02/24/2000 15:...
☑	5413cover.fm	383.0...	02/24/2000 15:...	02/24/2000 15:...
☑	5413edno.backup.fm	27.0...	02/24/2000 15:...	02/24/2000 15:...
☑	5413edno.fm	27.0...	02/24/2000 15:...	02/24/2000 15:...
☑	5413eva.backup.fm	26.0...	02/24/2000 15:...	02/24/2000 15:...
☑	5413eva.fm	26.0...	02/24/2000 15:...	02/24/2000 15:...
☑	5413glos.fm	13.0...	06/14/1999 13:...	02/24/2000 15:...

Displaying \\a78awzzb\c$\cbak

For Help, press F1

Figure 13. Windows NT interface showing collapsible directories

When restoring, only the files that have been backed up are displayed. Figure 14 shows an example of a restore window with the files in a directory available to restore.



- ANODE
  - Backup Sets
  - File Level
    - \\a78awzzb\c$ (C:)
      - □ Acrobat3
      - □ ADOBEAPP
      - □ adsm.sys
      - □ anno
      - ☑ hns
      - □ dsmdb
      - □ ExecSoft
      - □ fmtest
      - □ hbak
      - □ lotus
      - □ pdf
      - □ Program Files
      - □ PSFONTS
      - □ RECYCLER
      - □ sdwork

Name	Size	Modified	Created
☑ 5413.backup.book	31.0...	11/29/1999 08:...	02/24/2000 15:...
☑ 5413.pdf	4.08...	02/09/2000 13:...	02/24/2000 15:...
☑ 5413.ps	21.1...	11/29/1999 13:...	02/24/2000 15:...
☑ 5413abrv.fm	14.0...	06/14/1999 13:...	02/24/2000 15:...
☑ 5413ax01.fm	20.0...	06/14/1999 12:...	02/24/2000 15:...
☑ 5413ax02.fm	20.0...	06/14/1999 12:...	02/24/2000 15:...
☑ 5413base.prn	20.6...	12/14/1999 11:...	02/24/2000 15:...
☑ 5413base.ps	21.0...	12/14/1999 13:...	02/24/2000 15:...
☑ 5413bibl.backup.fm	46.0...	02/24/2000 15:...	02/24/2000 15:...
☑ 5413bibl.fm	46.0...	02/24/2000 15:...	02/24/2000 15:...
☑ 5413biblold.fm	47.0...	11/23/1999 08:...	02/24/2000 15:...
☑ 5413cover.backup.fm	383.0...	02/24/2000 15:...	02/24/2000 15:...
☑ 5413cover.fm	383.0...	02/24/2000 15:...	02/24/2000 15:...
☑ 5413edno.backup.fm	27.0...	02/24/2000 15:...	02/24/2000 15:...
☑ 5413edno.fm	27.0...	02/24/2000 15:...	02/24/2000 15:...
☑ 5413eva.backup.fm	26.0...	02/24/2000 15:...	02/24/2000 15:...
☑ 5413eva.fm	26.0...	02/24/2000 15:...	02/24/2000 15:...
☑ 5413glos.fm	13.0	06/14/1999 13:	02/24/2000 15:

Displaying \\a78awzzb\c$\cbak

For Help, press F1

Figure 14. Windows NT interface showing directories available to restore

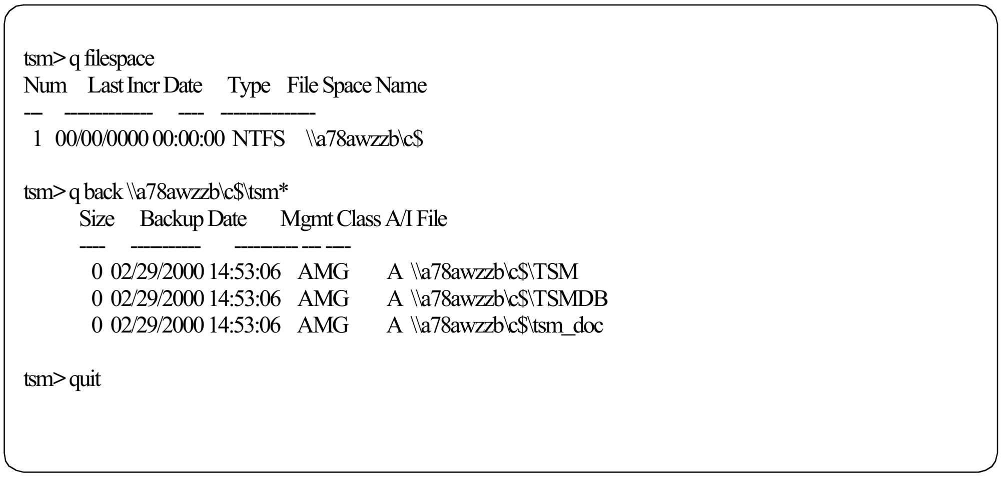
2.2.1.2 CLI

The command line interface (CLI) has a richer set of functions than the GUI. The CLI has the benefit of being a character mode interface, and therefore is well suited for those users who need to type the commands. You may also consider using it when you cannot access the GUI interface or when you want to automate a backup process by using a batch processing file. You may use it for backup, restore, archive, and retrieve operations, and to start the Tivoli Storage Manager scheduler.

There are two different methods to use the command line interface: interactive and non-interactive. The DSMC program activates both methods.

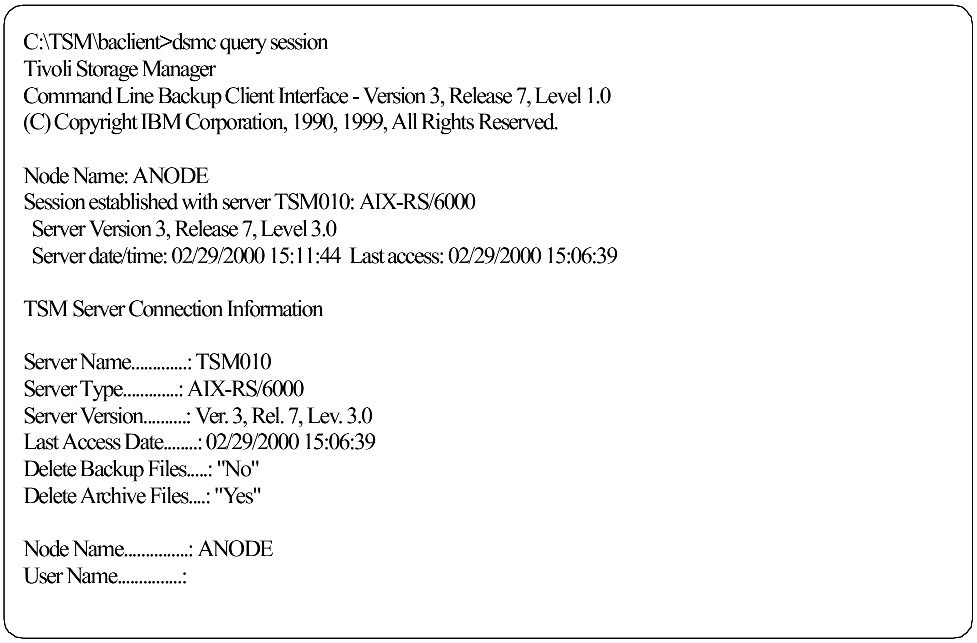
The interactive method (also called loop mode) has a prompt (dsmc>) to type all Tivoli Storage Manager specific commands for the client. If you use an interactive command line session, it is not necessary to precede each command with dsmc. If you use interactive mode you do not have to enter your password with each command.

The following example shows an interactive command line prompt and some backups already made. When you are running from the command line in this way, you must use the QUIT command to exit. Otherwise, DSMC waits for the next input command:



```
tsm> q filespace
Num    Last Incr Date      Type    File Space Name
---    ----------------    ----    ---------------
  1   00/00/0000 00:00:00  NTFS    \\a78awzzb\c$

tsm> q back \\a78awzzb\c$\tsm*
          Size     Backup Date        Mgmt Class A/I File
          ----     -----------        ---------- --- ----
            0  02/29/2000 14:53:06    AMG        A  \\a78awzzb\c$\TSM
            0  02/29/2000 14:53:06    AMG        A  \\a78awzzb\c$\TSMDB
            0  02/29/2000 14:53:06    AMG        A  \\a78awzzb\c$\tsm_doc

tsm> quit
```

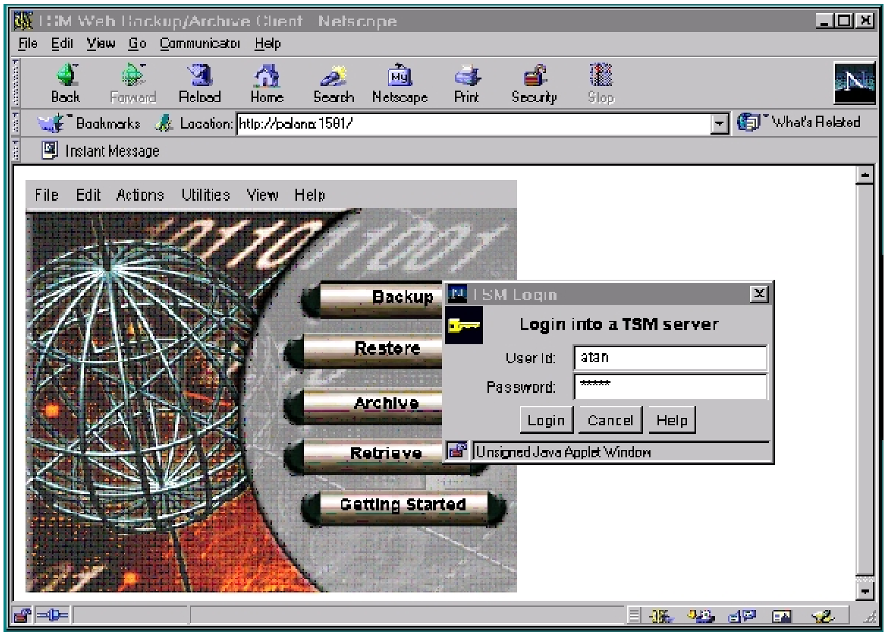
The non-interactive method (also called batch mode) requires that you use the dsmc and the actual command that you want to execute. In this mode, DSMC will process the command and return to the calling program. You can have a number of DSMC commands in a batch file for automatic processing. For example, to perform a Backup-Archive client QUERY operation and return to the operating system prompt, you should run the DSMC QUERY SESSION command:



```
C:\TSM\baclient>dsmc query session
Tivoli Storage Manager
Command Line Backup Client Interface - Version 3, Release 7, Level 1.0
(C) Copyright IBM Corporation, 1990, 1999, All Rights Reserved.

Node Name: ANODE
Session established with server TSM010: AIX-RS/6000
  Server Version 3, Release 7, Level 3.0
  Server date/time: 02/29/2000 15:11:44  Last access: 02/29/2000 15:06:39

TSM Server Connection Information

Server Name.............: TSM010
Server Type.............: AIX-RS/6000
Server Version..........: Ver. 3, Rel. 7, Lev. 3.0
Last Access Date........: 02/29/2000 15:06:39
Delete Backup Files.....: "No"
Delete Archive Files....: "Yes"

Node Name...............: ANODE
User Name...............:
```

2.2.1.3 Web client

The Web client permits an authorized administrator, helpdesk or end-user to perform backup and restore services on any machine that supports a Java 1.1.6-capable browser, such as Netscape 4.06 and Microsoft Internet Explorer 4.01 or higher. Multiple Web client sessions can be started simultaneously to perform different functions on separate Tivoli Storage Manager clients. Figure 15 shows you the main Web client screen and the login dialog box after clicking on **Backup**.


TSM Web Backup/Archive Client - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Location: http://palana:1581/ What's Related
Instant Message
File Edit Actions Utilities View Help
Backup
Restore
Archive
Retrieve
Getting Started
TSM Login
Login into a TSM server
Login Cancel Help
Unsigned Java Applet Window

User Id: atan

Password: *****

Figure 15. Web Backup-Archive client during authentication

2.2.2 Configuration and options files

Configuration files and option files are used to specify one or more servers and communication options for backup and restore services. The file can include authorization options, backup and archive processing options, scheduling options and where applicable, Tivoli Space Manager options.

On the UNIX platform, the Tivoli Storage Manager options reside in two separate options files, the client system options file (dsm.sys) and the client options file (dsm.opt). On other platforms, the options file (dsm.opt) contains all of the options. The end user sets up these files when the Tivoli Storage Manager Backup-Archive client is first installed on the user's workstation. Figure 16 shows a schematic view of the options file for the client.

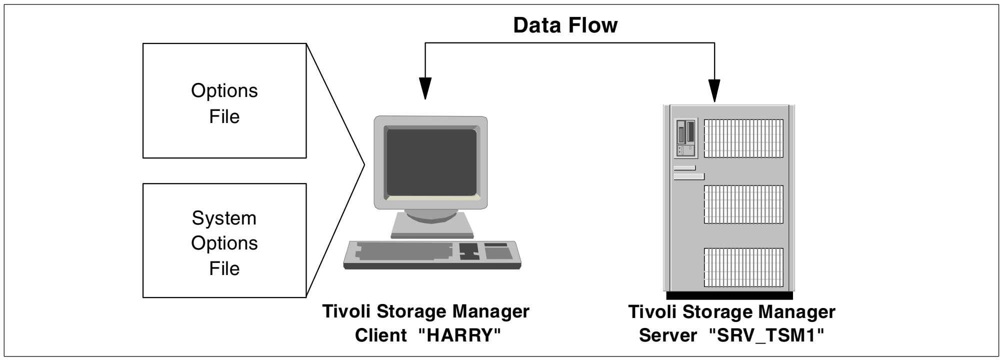

Data Flow
Options File
System Options File
Tivoli Storage Manager Client "HARRY"
Tivoli Storage Manager Server "SRV_TSM1"

Figure 16. Tivoli Storage Manager client with configuration and options files

The minimum configuration parameters for successful communication are:

- Communication Protocol—Client and server using the same type

- Tivoli Storage Manager Server address—Identifying the correct Tivoli Storage Manager server to use
- Nodename—The name by which the Tivoli Storage Manager server knows the client. This information is required so that the server allows access for this client. The node name and password are set up on the server, and if different from the machine name, they need to be added in the client options file also.

Figure 17 shows an example of a client (HARRY) and its minimum configuration settings.

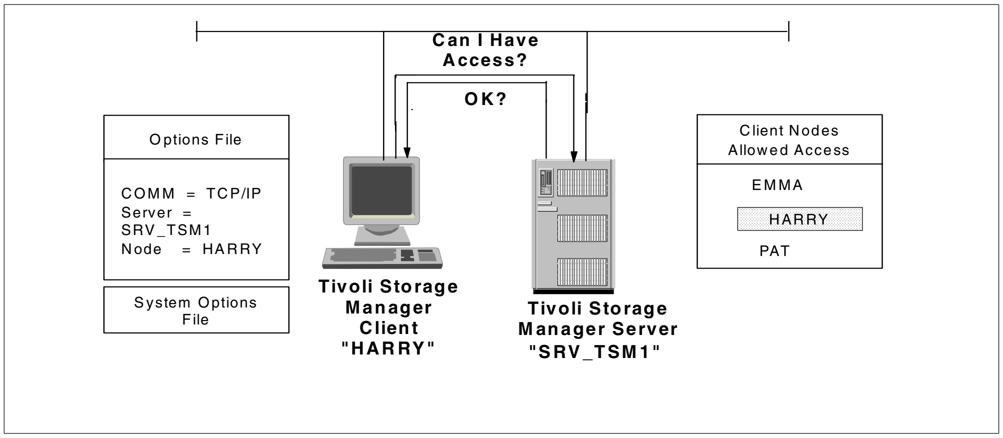


Figure 17. Client presents node name and is accepted

2.2.3 Establishing the session

The Tivoli Storage Manager Backup-Archive client node is required to register with the Tivoli Storage Manager server. Once registered, the Tivoli Storage Manager client can start its communication with the server by first completing a sign-on process. This sign-on process requires the use of a password that, when coupled with the node name of the client, insures proper authorization when it connects to the server. Figure 18 shows an example of the steps to establish a session with the Tivoli Storage Manager server.

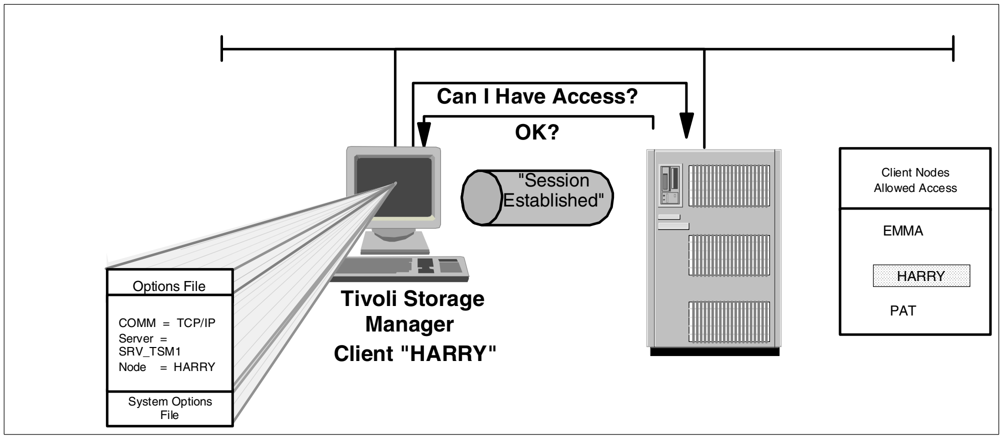


Figure 18. Establishing a client session

2.3 Tivoli Storage Manager multi-session and transaction concepts

The Tivoli Storage Manager clients use various internal techniques to improve performance. In this section we describe the multi-session capabilities of V3.7 and later clients and client transaction concepts.

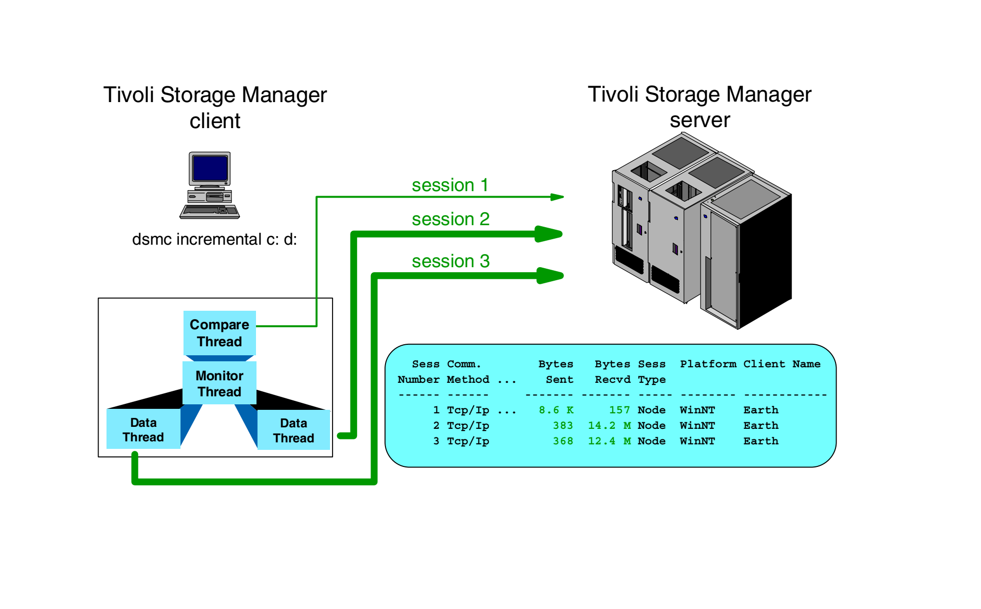
2.3.1 Multi-session

Tivoli Storage Manager exploits the multi-threading capabilities of modern operating systems by transparently initiating multiple backup or archive sessions on the client where necessary for rapid processing and data transfers between the client and the server.

The multi-session function involves three types of threads: compare, data transfer, and monitor. Compare and data threads show up as individual client sessions on the Tivoli Storage Manager server, while a monitor thread is not visible as a session.

- **Compare thread:** The purpose of the compare session is to generate a candidate list of files that have to be transferred to the Tivoli Storage manager server. As soon as one transaction limit is reached (see 2.3.2, “Transaction” on page 35), the list with the files for that one transaction will be handed over to a data transfer thread. There can be one or more compare threads at the same time.
- **Data transfer thread:** A data transfer thread is responsible for reading the files, compressing them (if required) and sending the data and waiting for the commit from the server. There can be one or more data transfer threads at the same time.
- **Monitor thread:** The task of this thread is to monitor if the overall backup or archive operation could benefit by starting more data sessions. Therefore, it is the monitor thread which decides on how many sessions are used.

In Figure 19 we can see how multiple sessions are started from the client to the server, and how they show up as individual sessions in a server query.



```
  Sess Comm.           Bytes    Bytes Sess  Platform Client Name
Number Method ...       Sent    Recvd Type
------ ------        ------- -------- ----- -------- -----------
     1 Tcp/Ip ...    8.6 K      157 Node  WinNT    Earth
     2 Tcp/Ip          383   14.2 M Node  WinNT    Earth
     3 Tcp/Ip          368   12.4 M Node  WinNT    Earth
```

Figure 19. Multi-session client

The administrator and the user each have controls to influence the number of sessions that a client can start. On the server, the global setting MAXSESSIONS limits the total number of sessions of any kind which may be present. The client node setting MAXNUMMP, in its server definition, controls how many mount points (for sequential devices such as tape drives) a client may allocate. Finally, the RESOURCEUTILIZATION setting in the client option file increases or decreases the ability of the client to create multiple sessions.

2.3.2 Transaction

All data sent to Tivoli Storage Manager storage during a backup or archive session is done within the bounds of a transaction. When the client starts sending or receiving data, it pays attention to both sides of the communication (the server and the client). All operations are controlled in a such way that Tivoli Storage Manager can detect any data inconsistency during the transfer (due to a network problem, hard drive full or a file that already exists, for example). This provides a high level of data integrity for the Tivoli Storage Manager product. A single transaction is an atomic action, the smallest possible unit of work. Data sent within the bounds of a transaction is either committed completely to the system at the end of the transaction, or it is all rolled back if the transaction is ended prematurely.

Figure 20 shows an example of two backup transactions (Transaction 001 and Transaction 002) that are on their way to the server. Neither of them have finished yet, because the client is still sending the files related to those transactions. As the server receives the files, it saves them in storage, but it will only commit the transaction in the server database after receiving all the associated files.

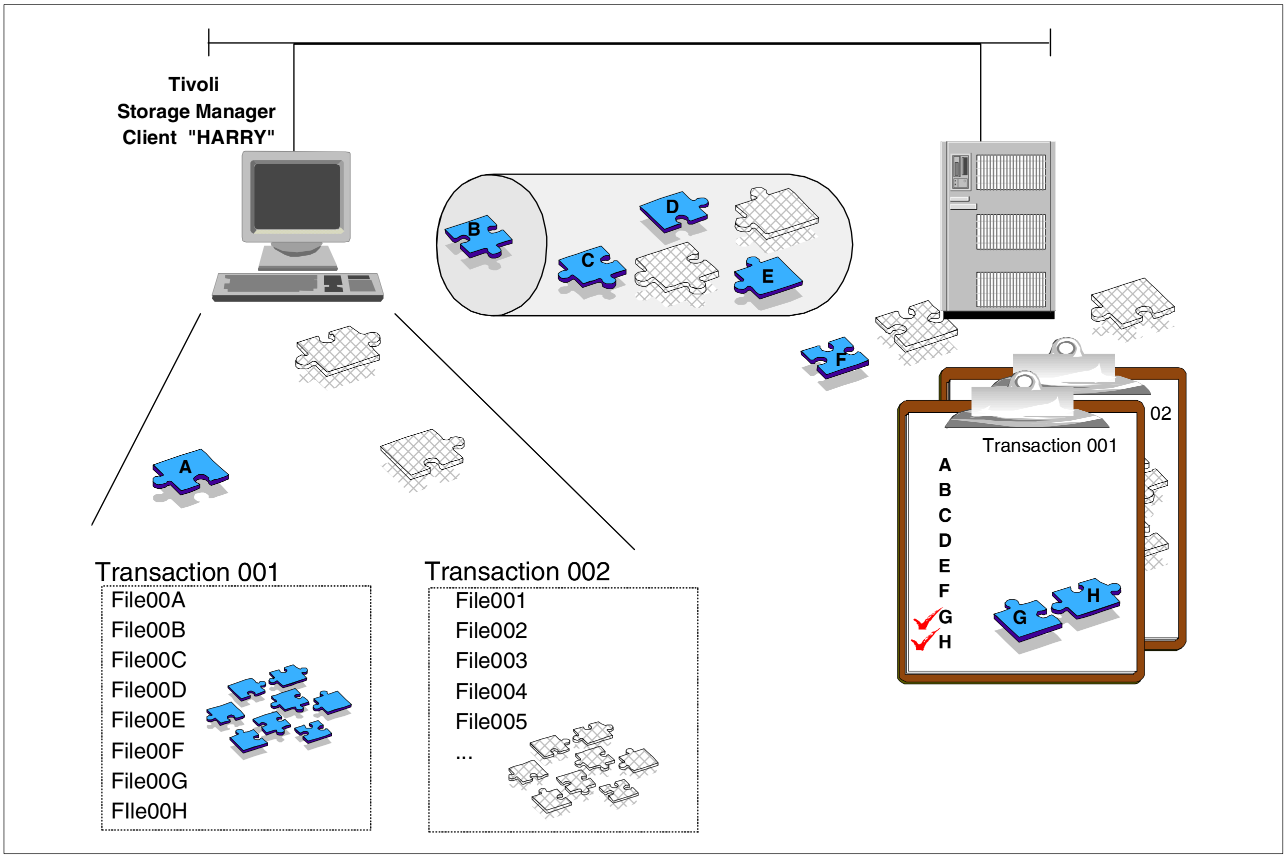


Figure 20. Transaction processing

The size of a transaction is controlled by the server setting TXNGROUPMAX, which sets the maximum number of client files which can comprise a single transaction, and the client setting TXNBYTELIMIT which sets the maximum number of bytes that can be sent. Whichever limit is hit first as the client is sending its files, will determine the complete transaction.

2.4 Backup

Tivoli Storage Manager can perform backups of both files and raw logical volumes. When backing up files, the Tivoli Storage Manager server database keeps a list of all files and their attributes (time, date, size, access control lists, extended attributes). At each file backup operation, this list is compared to the current file system on the client workstation to determine new, deleted and changed files. Raw logical volumes are treated as separate entities and the management class policy is applied to the entire image as a whole. There is no tracking of individual files in an image backup, that is, it is treated as a separate object. More details on image or raw logical volume backup are given in 2.6, “Logical volume backup” on page 48.

During backup, the client first establishes a session with the Tivoli Storage Manager server. After that, it sends the data using the transaction controls as explained in 2.3.2, “Transaction” on page 35.

Tivoli Storage Manager a number of backup versions for each file or object on each client node. If and when the number of versions stored on the server exceeds the number set by the Tivoli Storage Manager administrator, older versions are deleted as newer versions are made. When you back up files, Tivoli Storage Manager also backs up all related directory information and access information. (See Figure 21.)

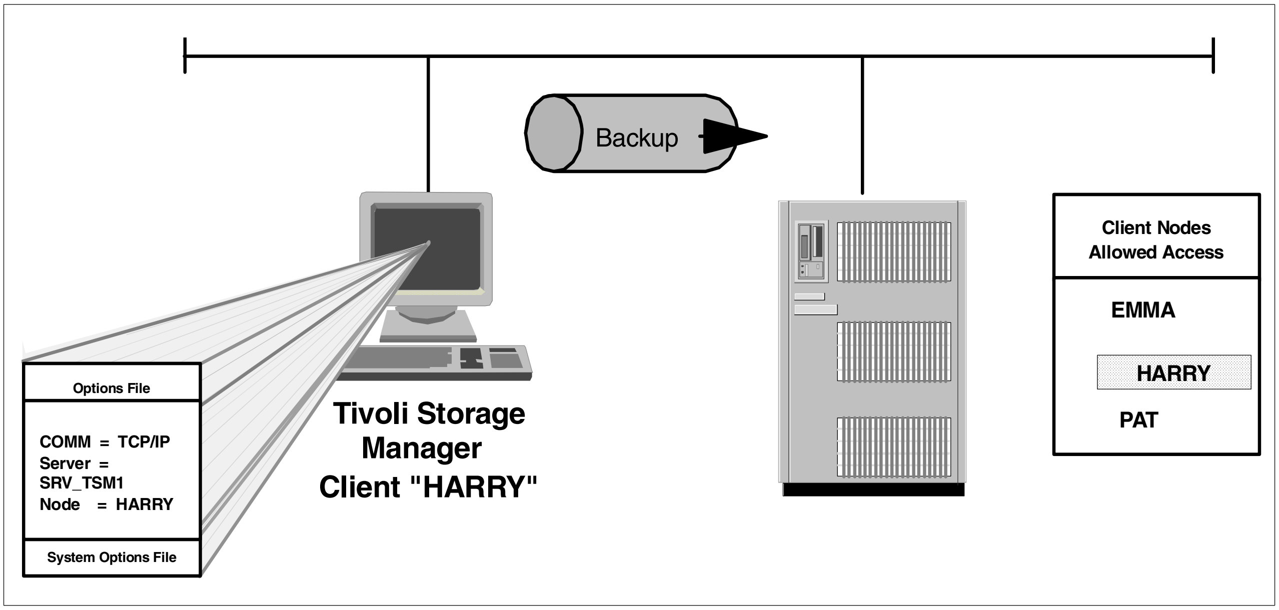


Figure 21. Backup in progress

2.4.1 Backup key concepts

There are two types of file backup: incremental and selective. An incremental backup backs up files, directories, or subdirectories that are new, or that have changed since the last incremental backup. A selective backup backs up specific files, or entire directories unconditionally.

2.4.2 Incremental backup

Tivoli Storage Manager is unique in offering an *incremental* or *progressive backup* methodology for backing up client data. This approach can remove the need for periodic full dumps, since only the changed files are backed up. This can have significant benefits in backup time, number of tapes used, reduced network traffic, size of backup servers and manageability.

The *incremental* backup operation is a full scan of the client’s filesystems which backs up all files and other information necessary (and only those things) to make sure that the Tivoli Storage Manager inventory matches the current state of the client’s storage. This means that the first time this is run on a new client, everything is backed up. Each time after this, only new and changed files are sent. During the incremental backup, the client queries the Tivoli Storage Manager server so that it knows what files are currently stored. The client uses this information to:

- Back up new files.
- Back up files whose contents have changed.

- Expire backup versions on the server for files that were deleted from the workstation.

Figure 22 shows an example of a daily incremental backup. On DAY1, we have two files (fileA and fileB). On DAY2, fileC is first written, and therefore, this is when it is first backed up. When we move to DAY3, fileB and fileC are the only files copied again, because they are the only ones that have changed since the last backup.

In our example, fileA has never changed, and therefore, Tivoli Storage Manager only stores one copy of this file. However, the changed files, fileB and fileC, have two copies stored in the server.

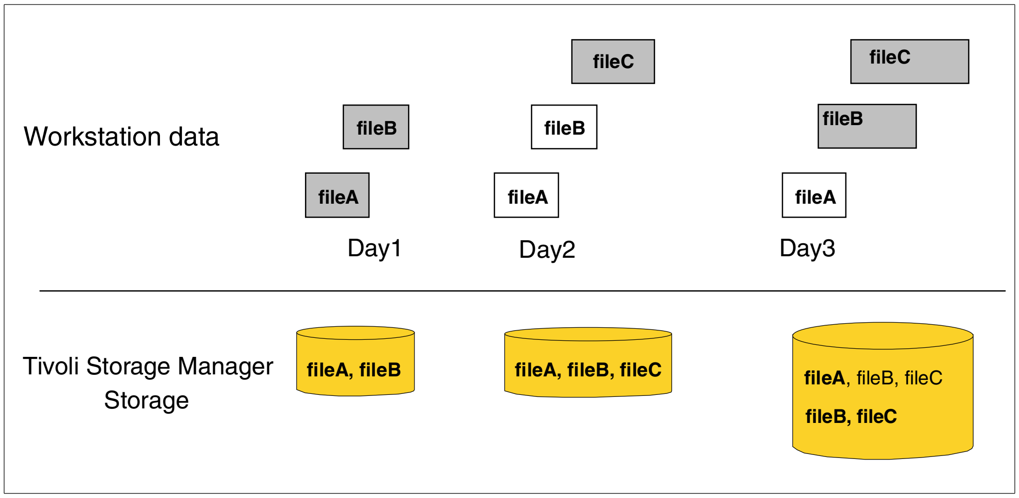


Figure 22. Incremental backup processing

2.4.3 Selective backup

During a selective backup, Tivoli Storage Manager sends copies of the files to the server even if they have not changed since the last backup. This is useful if you want to back up many files that are not in the same directory structure, regardless of their actual status in the Tivoli Storage Manager server. It may also apply where you want to enforce a complete backup.

However, remember versioning still applies. If you back up a file multiple times when it has not changed, this will result in having multiple copies of exactly the same file on the server, instead of a number of different versions of the file. This more or less defeats the purpose of Tivoli Storage Manager version control. To avoid that, use the incremental backup technique command to back up only changed and new files. Typically Selective backup will only be used in special circumstances.

Figure 23 shows an example of a selective backup. On DAY1, fileA and fileB are first backed up. Because we are running the selective command, they are also backed up, even though they are unchanged, on DAY2, together with fileC which is new. The same thing happens on DAY3: fileA, fileB, and fileC are backed up again to the Tivoli Storage Manager storage.

In the example, our final storage holds three identical copies of fileA, three copies of fileB (two identical) and two different versions of fileC.

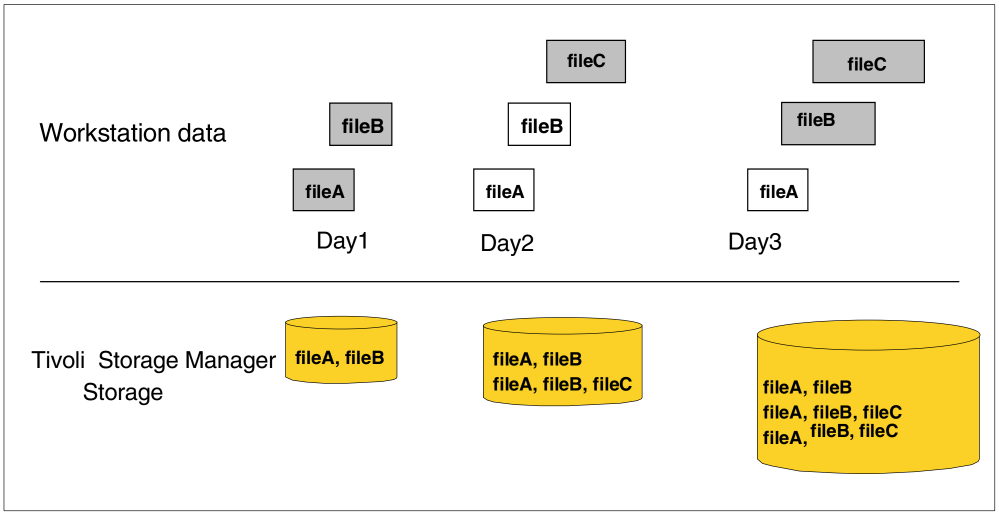


Figure 23. Selective backup processing

2.4.4 Include-exclude lists

Tivoli Storage Manager gives a high level of control over which client data are actually backed up and which are excluded. It also controls the management class which is used for those files which do get backed up. The mechanism for controlling this on the client side is called the include/exclude list. Management classes are defined by the Tivoli Storage Manager administrator and are described in more detail in Chapter 3, “Data storage policy” on page 63.

If you do not specify any control, all locally attached client files and directories will be backed up and will be bound to a DEFAULT management class. However, this may take up much space in server storage, or the backup operation may take too long to complete. You may only need to backup some particular drives or filesystems, excluding all others. Or, you may want to exclude from backup any files with a .TMP extension. You may need different management policy for certain critical files (such as spreadsheets, documents, and e-mail messages), and for ordinary files that you can easily get back if necessary (such as Internet files and temporary files). Tivoli Storage Manager allows you to control how files are stored, with respect to their management class. The include/exclude list is a set of references local to each client which controls what files are and are not backed up, and what management policy is used. If you do not select any special management, Tivoli Storage Manager will use a DEFAULT management class.

Figure 24 shows how the Backup-Archive client decides what files are stored in the server, and how they are stored. The INCLUDE rules explicitly tell how to handle data. The EXCLUDE rule forces Tivoli Storage Manager to skip those objects. Everything else that is not defined in these rules is INCLUDED by DEFAULT.

In the example, files from the /home directory are bound to the PERSONAL management class. All files from the /public directory are bound to the SHARED management class. Because the include-exclude list does not reference any other special management class for files, all the others are directly bound to the DEFAULT management class in the server, except the /temp directory, which is prevented from being backed up.

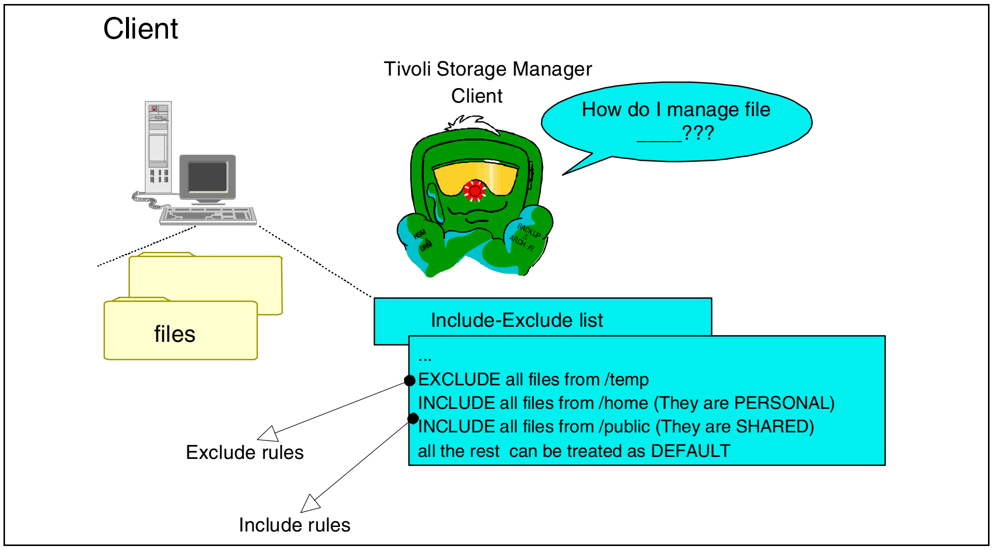

Client
Tivoli Storage Manager Client
How do I manage file ____???
files
Include-Exclude list
...
EXCLUDE all files from /temp
INCLUDE all files from /home (They are PERSONAL)
INCLUDE all files from /public (They are SHARED)
all the rest can be treated as DEFAULT
Exclude rules
Include rules

Figure 24. Include-exclude list and rules

Include-exclude rules are checked from the bottom up until a match is found. If a match is found, the processing stops and checks whether the option is include or exclude. If the option is INCLUDE, the file is backed up to either a specified management class or the DEFAULT, if omitted. If the option is exclude, the file is not backed up. Any files which do not match any rule are backed up to the DEFAULT management class.

Suppose that you have the following include-exclude rules:

(3) include /temp/filedir/*

(2) exclude /temp/filedir/project/*

(1) include /temp/work/*

Because Tivoli Storage Manager performs include-exclude processing from bottom-up, it starts validating rule (1) through rule (3). Therefore, if we have a file called /temp/work/file1, it is included by rule (1). On the other hand, if we have a file called /temp/filedir/project/newfile, this one is excluded by rule (2). A file called /temp/filedir/otherfile is included by rule (3).

2.4.5 Active and inactive file versions

One of the most important concepts in Tivoli Storage Manager data management is the difference between an active backup version and an inactive backup version.

Assume a new file is created on your workstation. The next time you run a backup operation (say Monday at 9pm), Tivoli Storage Manager server stores this file. This copy of the file is known as the ACTIVE version. When you run an incremental backup again, (say Tuesday at 9pm) Tivoli Storage Manager uses this ACTIVE version already stored to check back with your workstation to determine if the file has changed since the last backup. If it has, it is backed up again. This version now becomes the ACTIVE version and the copy from Monday at 9pm becomes an INACTIVE. The most recent backed up version of the file is always the ACTIVE version, so long as it still exists on the original client. Tivoli

Storage Manager will keep storing a new ACTIVE version and inactivating the previous active version, up to the limit of the total number of versions defined to be retained in the management class. Once this limit is exceeded, the oldest INACTIVE version is deleted from Tivoli Storage Manager storage, and will no longer be able to be restored. This process of maintaining the ACTIVE and INACTIVE versions (up to the management class limit) continues until the file is deleted from the original client. If this occurs, when the next incremental backup is run, the server detects that the file no longer exists on the client. All stored versions of the file now automatically become INACTIVE, and some of the oldest versions of the file may also be deleted. This would happen if the management class setting for number of versions to retain of a deleted file is less than the number of versions defined to retain of an existing file. Therefore, an ACTIVE file version is stored, and also INACTIVE versions as long as the file is still resident on the client system. If the file is deleted, then only INACTIVE version(s) of the file will exist in server storage. The absence of an ACTIVE file version in Tivoli Storage Manager storage indicates that the file has been deleted from the client machine, and therefore, the only copy is now in Tivoli Storage Manager storage.

Tivoli Storage Manager controls the retention of its ACTIVE and INACTIVE versions of a file that exists on a client machine by using two criteria defined in the Management Class:

- **How many versions:** The parameter that controls the number of backup versions is called VEREXIST. This may be set at a specific number or to UNLIMITED.
- **How long to keep:** The RETEXTRA parameter controls how much time must elapse before an INACTIVE file version is considered expired. This parameter controls how long to retain all remaining inactive files and may be set at a specific number of days or to NOLIMIT - meaning they will never be expired.

Important

An ACTIVE file version is never expired. Even if you never change a particular file after the first incremental backup, Tivoli Storage Manager will keep this file version indefinitely.

For a file deleted on a client machine, Tivoli Storage Manager uses different criteria:

- How many files: The parameter that controls the number of inactive backup versions is called VERDELETED. This number is normally less than or equal to the number you have for VEREXIST.
- How long to keep files: The RETEXTRA parameter controls how much time must elapse before an INACTIVE file version is considered expired. This parameter controls how long to retain all remaining inactive files, except for the last one and may be set at a specific number of days or to NOLIMIT — meaning they will never be expired.
- How long to retain the last file: The RETONLY parameter controls the last inactive copy of a file. As files get expired by RETEXTRA, you can configure Tivoli Storage Manager to manage the last inactive copy differently, so that you can keep that file for a longer period of time. It may be set at a specific number of days or to NOLIMIT — meaning they will never be expired.

- Typically, you configure RETONLY to be either the same value or longer than RETEXTRA, since it functions as a grace period before finally expiring the file.

Figure 25 shows an example of a file (file1) that was first backed up on January 1st and then on January 5th, 15th, 20th, and 22nd. On January 22nd, the backup procedure had four versions of the same file (January 22nd being the active and most recent copy, and the January 1st copy being expired due to VEREXIST limits).

When file1 is deleted on the client on January 23rd and expiration runs, all file1 versions become inactive and the Tivoli Storage Manager server now uses the VERDELETED information to reduce the number of inactive files. Because VERDELETED is set as 2, old versions of the file are immediately expired. Now, file1 has only two versions in the Tivoli Storage Manager server (20/Jan and 22/Jan). These files are now handled by RETEXTRA until their expiration value is reached.

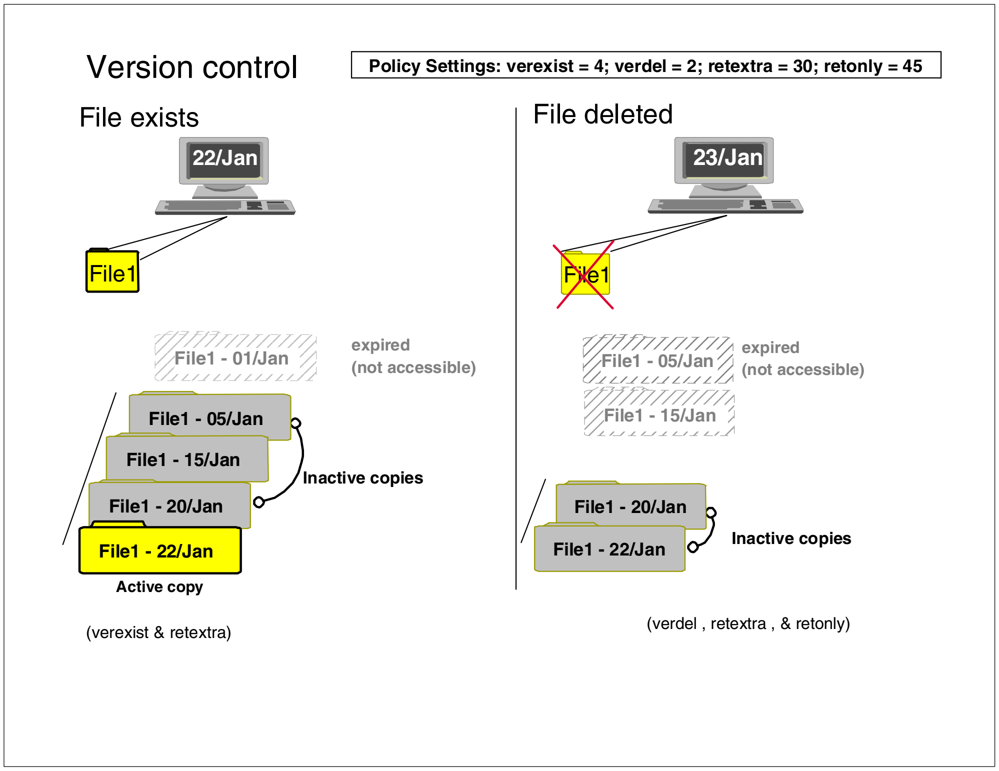
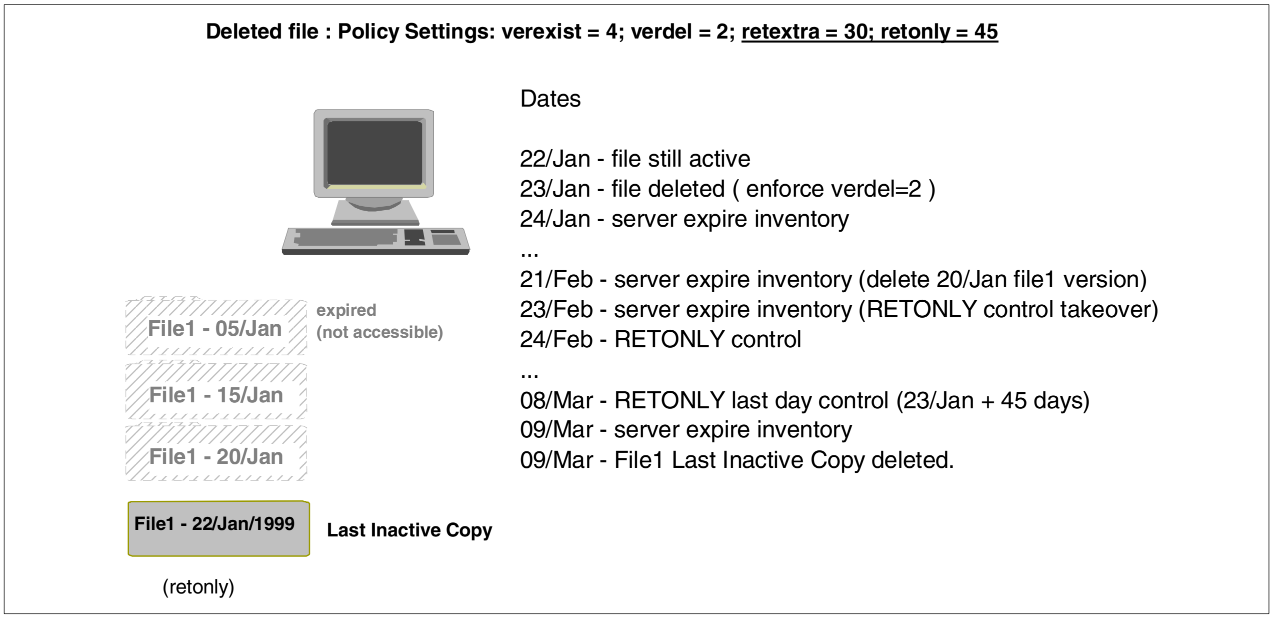


Figure 25. Active and inactive files

2.4.6 Retention

The retention period of a file version is the length of time which that file is maintained by Tivoli Storage Manager and accordingly is available to be restored to the client. When a file version is no longer retained, then it is expired from the Tivoli Storage Manager database. A file version is expired either because it is superseded by version control (VEREXISTS, VERDELETED) or it is older than the retention period (RETEXTRA, RETONLY). Retention only applies to INACTIVE files because ACTIVE files are never expired. The retention period is measured from the time when the file version becomes inactive.

In our example, Figure 26 shows a scenario in which the last inactive backup copy of file1 will be kept up to March 9th, 2000.



Deleted file : Policy Settings: verexist = 4; verdel = 2; retextra = 30; retonly = 45

Dates

22/Jan - file still active
23/Jan - file deleted (enforce verdel=2)
24/Jan - server expire inventory
...
21/Feb - server expire inventory (delete 20/Jan file1 version)
23/Feb - server expire inventory (RETONLY control takeover)
24/Feb - RETONLY control
...
08/Mar - RETONLY last day control (23/Jan + 45 days)
09/Mar - server expire inventory
09/Mar - File1 Last Inactive Copy deleted.

Figure 26. RETEXTRA and RETONLY expiration processing

As a rule of thumb, VEREXIST should be greater than or equal to VERDELETED (VEREXIST >= VERDELETED) and RETONLY should be greater than or equal to RETEXTRA (RETONLY >= RETEXTRA).

2.4.7 Backup binding

Any object stored in the Tivoli Storage Manager server is controlled by specific criteria in such a way that the server knows how long that object must be kept. In other words, a backup file, directory and any other object stored in Tivoli Storage Manager has one management class reference that dictates how many copies can be kept, and how long a backup file must exist in the server. The link process between the file and the management class is called binding.

The binding process occurs when you perform an incremental backup. The Tivoli Storage Manager client checks both the server management classes and the client include-exclude list or client option files to perform the binding process.

Figure 27 shows an example of many files and their management class bindings. In the example, files from the /home directory are bound to the PERSONAL management class. All files from the /public directory are bound to the SHARED management class. Because the include-exclude list does not reference any other specification for files, our /adsm directory is implicitly bound to the assigned default management class in the server. An object can only ever be bound to one management class.

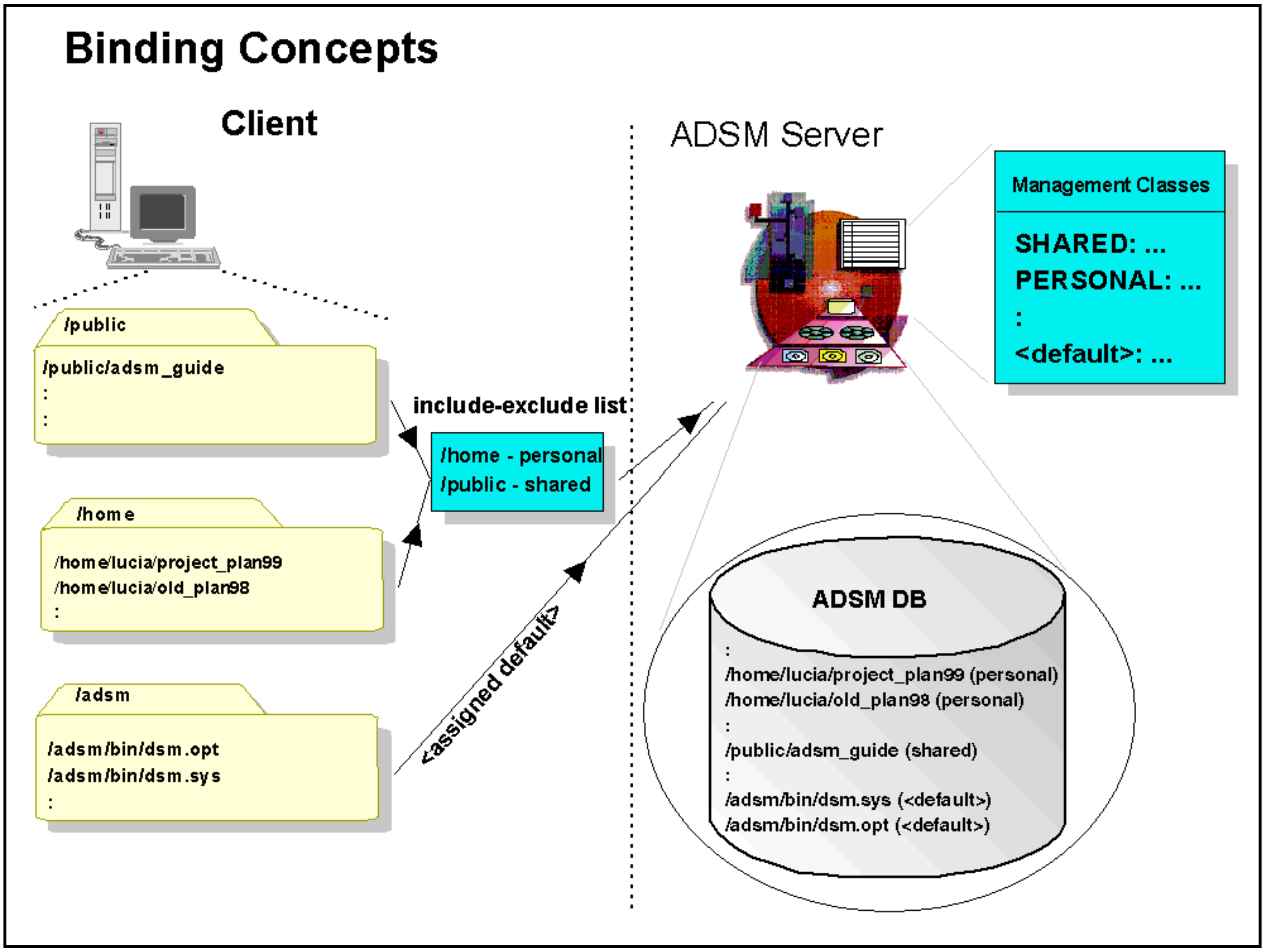


Figure 27. Binding files to management classes

2.4.8 Rebinding

Because a file stays bound to one single management class, every time you run a new incremental backup operation, the server checks to see if the file still belongs to the same management class and changes the class if necessary. This process is transparent to the user and is called rebinding the file. A file or directory will be rebound in the following cases:

- **Include-exclude list:** If you change the management class for a file, this affects all previous backup versions of that file and all future backup operations for that file as well.
- **Changes in the number of management classes:** If you delete or rename a management class, the server rebinds the files to either a DEFAULT class or uses grace periods in the policy domain to control retention. In this case, the server controls data by retention and not by versions. This may not be desirable, since you do not have a management class controlling those files. Although the rebinding process for this case was unsuccessful, Tivoli Storage Manager still controls the file by other means.

For further details on management classes and settings, see Chapter 3, “Data storage policy” on page 63.

2.4.9 Backup special considerations

The Tivoli Storage Manager incremental backup actually offers two options, *complete* and *date only*. These are also known as *full* and *partial incrementals*, or *full incremental* and *incremental-by-date*. Normally you would use the default, which is the complete or full incremental. This is the option described in 2.4.2, “Incremental backup” on page 37. Many attributes of each file are examined to determine if it has changed and therefore needs to be backed up. In a partial incremental operation, the client only asks the server for the date and time of the last incremental backup. If a file’s last changed date and time is after that of the last backup, the file is backed up. Otherwise it is not, even if the file is new to the workstation. Because changes to ACL or file permissions do not change the last changed date and time attributes, a partial incremental would not detect this, so the file would not be backed up. A file which is copied or created on the client with an older date/time stamp than the date/time of the last backup would also not be sent. In a partial backup, deleted files from the client are not recognized, and no version expiry or rebinding takes place.

Because a partial incremental backup only performs a fraction of the processing of the normal full backup, it does not ensure that the files contained in the server storage exactly match the current state of the files on the client. For example, files that would normally be backed up during a full incremental might not be backed up during a partial incremental; and old files that should be deleted from the server might not be deleted. Since this is the case, why have the partial incremental possibility at all? The reason is that this operation takes less time to complete than the regular full backup. So, in some particular circumstances where there is a problem meeting a limited backup window, you might use a partial incremental operation. If you do this, you MUST remember to periodically run full incremental backups to bring the Tivoli Storage Manager server in line with your workstation's status. For example, if you have only a limited time during the week to perform backups, but have extra time on the weekend, you can use partial backups on the weekdays, and then use full incremental backups on the weekends. However, if there is not a problem with the backup window, then always run the full incremental option.

2.5 Archive

The Tivoli Storage Manager Archive function stores selected files unconditionally on the server, according to the applicable management class limits. Unconditionally means that there is no version limit and they will be retained for the defined time period regardless of whether they are deleted on the client. Archived files are useful if you want to take a snapshot of particular files, or if you want to delete files, freeing up space, yet still have the ability to retrieve them if required. It is very common to have a legislative requirement to archive business records for long periods of time, and the archive function is ideal for this purpose.

Figure 28 shows a schematic archive operation.

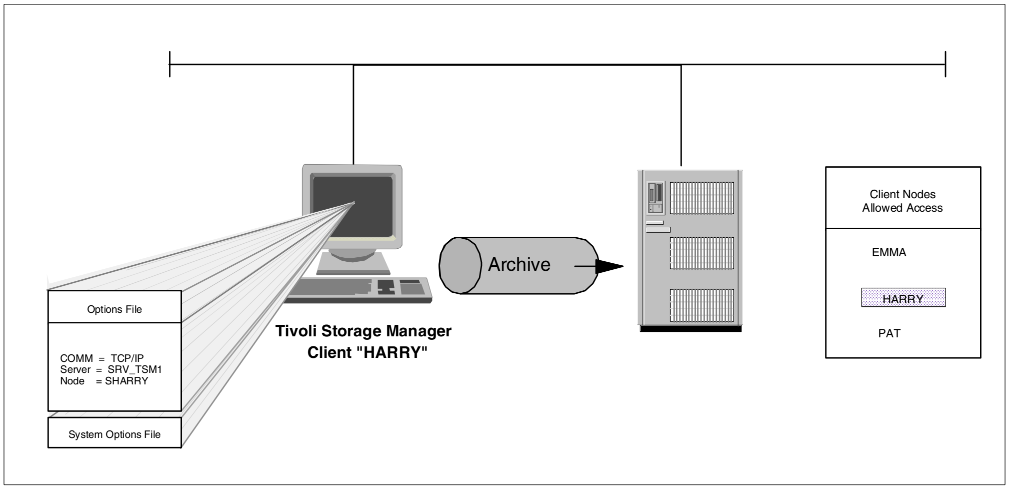


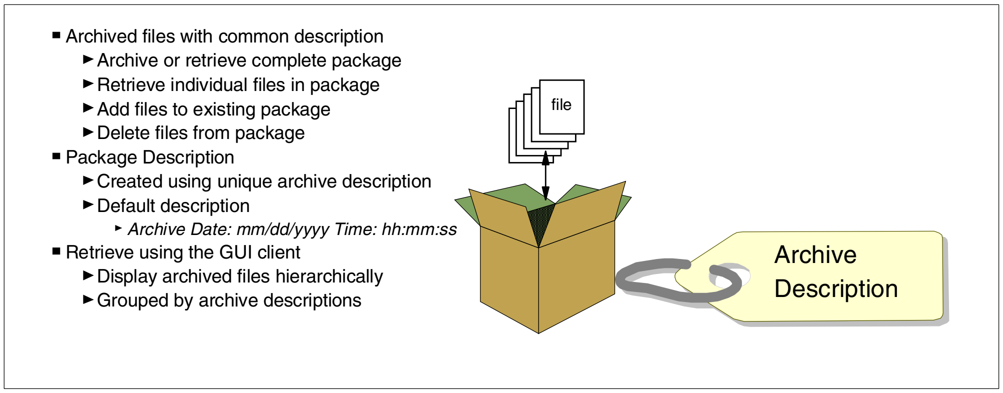
Figure 28. Archive in progress

2.5.1 Packages

Archive packages are groups of files archived together with a common description. The system will automatically supply a description consisting of the time and date stamp, or you can override this with your own meaningful description. This description is used later to allow easy searching and selection of archive packages to retrieve.

You can add to an existing package on a subsequent archive operation by supplying an existing archive package description.

You can either retrieve individual files within a package or delete files from a package. Figure 29 shows a summary of the packaging features.



- Archived files with common description
  - Archive or retrieve complete package
  - Retrieve individual files in package
  - Add files to existing package
  - Delete files from package
- Package Description
  - Created using unique archive description
  - Default description
    - *Archive Date: mm/dd/yyyy Time: hh:mm:ss*
- Retrieve using the GUI client
  - Display archived files hierarchically
  - Grouped by archive descriptions

Figure 29. Packaging files

2.5.2 Client space reduction

You can use archives to release storage space on your workstation and delete files as you archive them. This is useful when you know in advance that it is no longer necessary for a file to stay in the workstation, but it may still be important to keep the file for some time. You can archive and delete any file and use any retention period available in the archive management classes.

Figure 30 shows an example of an employee who has left the company. Because this employee had many files that may be needed, all files are saved for 30 days and deleted from the fileserver to make room for other users. The archive function includes an option to automatically delete the files after they are successfully received at the server.

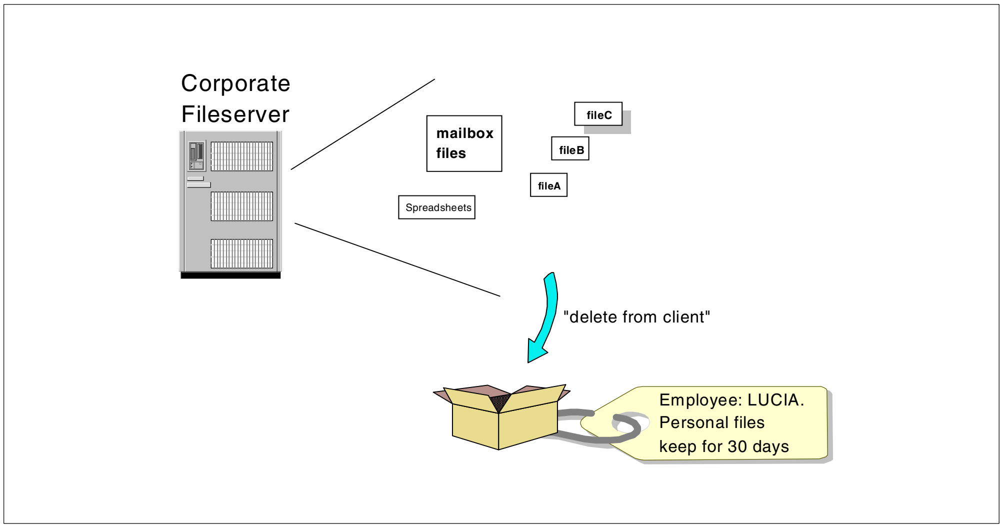


Figure 30. Archiving unnecessary files

2.5.3 Retention

When you archive a file, you can select between the available management classes that your client machine has access to. This makes it possible for you to select the retention period (according to the retention period in days specified in the archive copygroup) for all the data you are archiving. Archiving is different from file backup in that the user may select from the available management classes to determine the retention. For file backup, the user may not control the management class to which the files are bound.

Typically, you might use archives to save information that has either a legal requirement (such as account information, annual reports, billing information, and annual customer reports) as shown in Figure 31, or even an internal audit requirement (to keep application logs, user activity information, employee files, and so on).

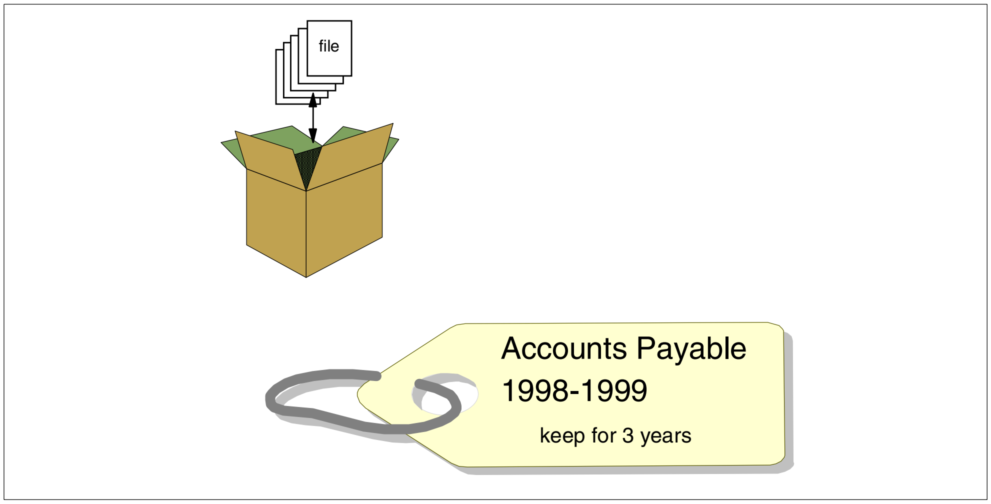


Figure 31. Archiving long-term files

2.6 Logical volume backup

Tivoli Storage Manager enables you to backup a file system or raw logical volume as a single object from your client machine. The Tivoli Storage Manager client accomplishes this by dynamically loading an image plug-in utility which sends the object to the server using the Tivoli Storage Manager API. This capability is currently available for the AIX, HP-UX and Solaris clients and can be used on a logical volume whether or not there is an associated file system. This will ensure a clean backup. Figure 32 shows the operation of the image backup and restore operation as a single object.

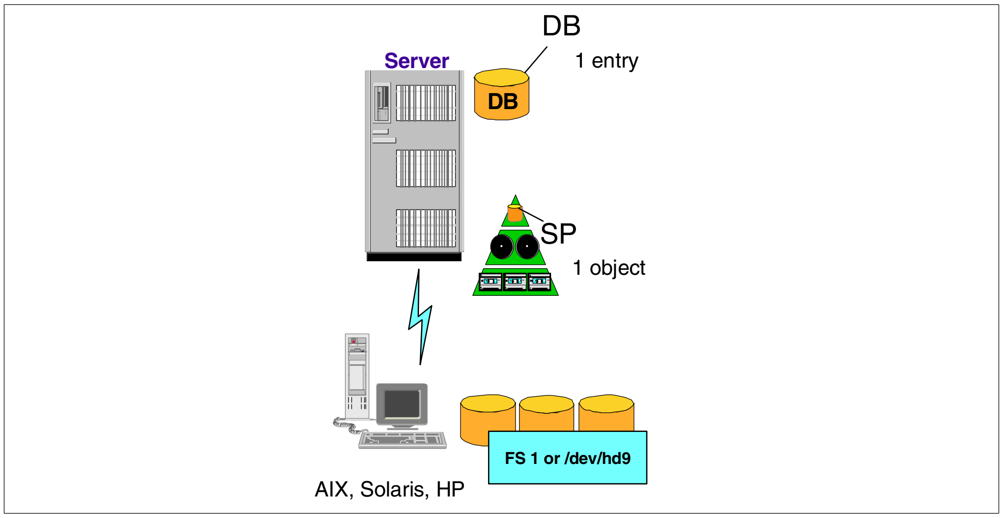


Figure 32. Image backup and restore

Logical volume backup has the advantages of improved backup and restore speeds, and conserving server resource since the entire backup is treated as a single object and individual files are not processed during backup or restore. Similar to the standard incremental and selective backup filtering options, you can include specific logical volumes and assign a Management Class to image objects and exclude file systems or raw device from being backup by specifying in the client include/exclude list as in the following examples:

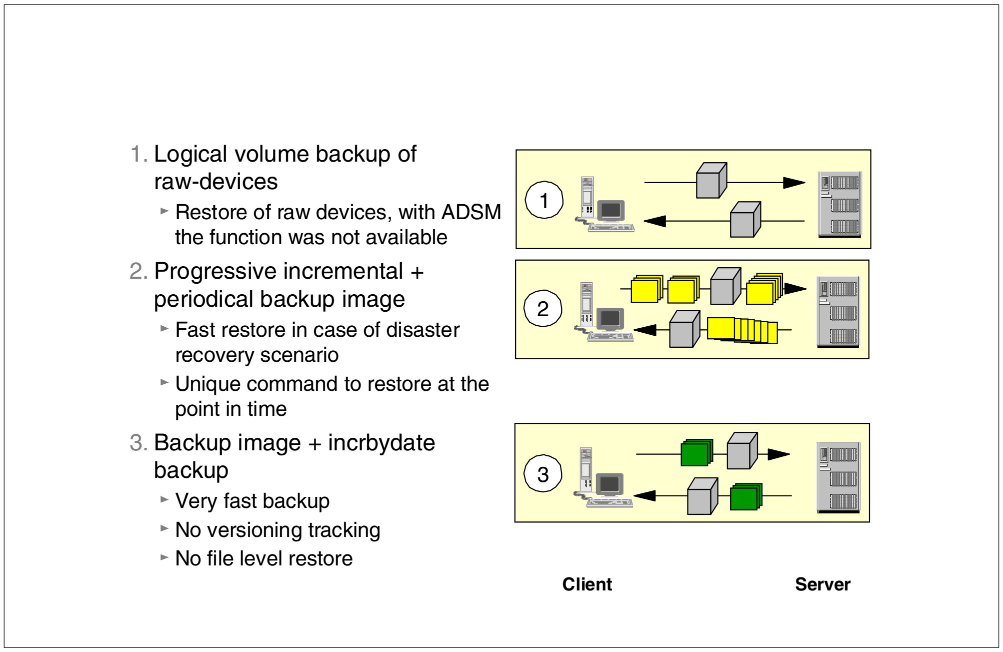
```
INCLUDE.IMAGE    /.../*    ImageMC

EXCLUDE.IMAGE    /dev/hd5
```

By setting the *Copy Serialization* parameter in your Management Class (see 3.2, “Copy groups” on page 64 for more information) as *Static* (or *Shared Static*), the filesystem (if one exists) will be unmounted first and remounted as read-only automatically before the backup proceeds. When the image backup is completed, the file system is remounted as it was originally. If the *Copy Serialization* parameter in your Management Class is *dynamic* (or *Shared Dynamic*), the client performs the backup of the file system even if it is in use. This is not recommended as it will probably lead to an inconsistent backup image. The automatic mounting and unmounting operations are not applicable for raw devices image backup since there is no associated filesystem.

You can consider full logical volume backup (mode=selective), on its own or in combination with progressive backup operations based on your requirement as in the following:

- Perform regular logical volume backup for raw devices used for application managed data spaces. Examples are offline backup of raw logical volumes used in database applications. Note that there is no incremental option in raw logical volume backup.
- Perform a combination of progressive and occasional image backup of your file system. This provides you with fast recovery as well as file level restore capability.
- Perform combination of periodic image backup with daily incremental-by-last-image-date backup. The MODE option in the backup image command determines whether the backup is a full file system image backup (*Selective*) or an incremental-by-last-image-date backup (*Incremental*). This provides you with fast backup and recovery of the entire filesystem. File level restore is only limited to changed files since the last full image backup. It is important to note that for incremental-by-last-image-date backup to work, you must not have any previous incremental backup of the filesystem using the `incremental` command.

These options are illustrated in Figure 33.



1. Logical volume backup of raw-devices
   - Restore of raw devices, with ADSM the function was not available
2. Progressive incremental + periodical backup image
   - Fast restore in case of disaster recovery scenario
   - Unique command to restore at the point in time
3. Backup image + incrbydate backup
   - Very fast backup
   - No versioning tracking
   - No file level restore

Figure 33. Options for image backup

Only a UNIX root user can perform the backup and restore image operations. When you restore from an image backup, the entire previous contents of the logical volume or filesystem will be overwritten.

2.7 Backup set

Tivoli Storage Manager enables you to generate a copy of your client’s most current backup from the Tivoli Storage Manager server onto sequential media.This is accomplished using `generate backupset` command, which copies all the active file versions of the fileset from server storage onto the media. This copy of the backup, also call backup set or portable backup, is self-contained and can be used for restoration of the client’s data independent of Tivoli Storage Manager from a locally attached device which can also read this media, for example, a CD-ROM. This technique provides Tivoli Storage Manager client rapid recovery with no server and network (LAN-free) dependency. You can also transfer the backup set from one server to another by generating the backup set on the source server, then transporting the backup set volume and defining it to the destination server, assuming both servers have the same media type. The same node name is required to be registered on both servers. See Figure 34.

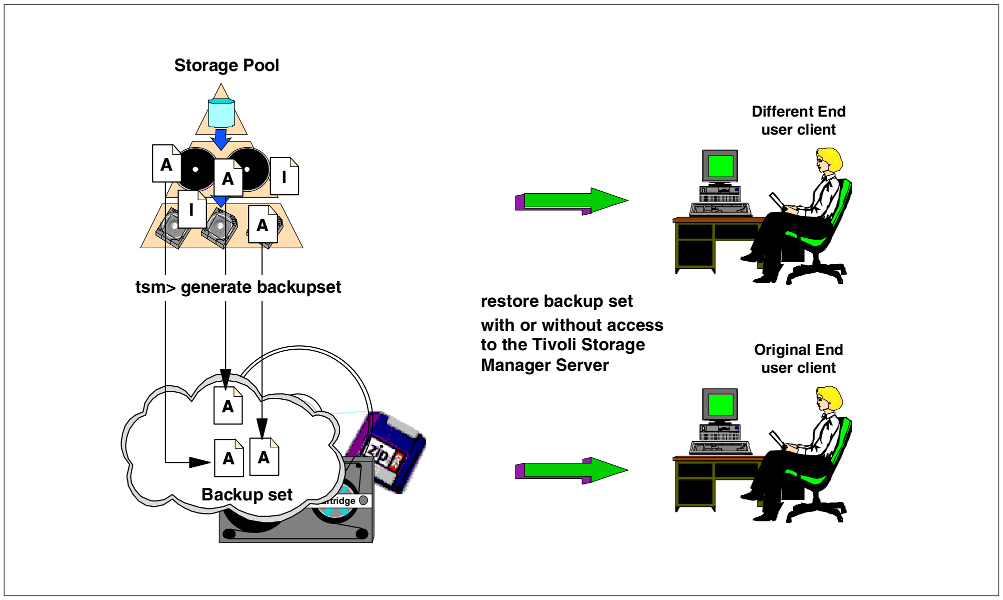


Figure 34. Portable client backup set

Note

LAN-free restore of a backup set still requires the Tivoli Storage Manager client to be installed. Therefore the portable backup set does not provide a 'bare metal restore' capability.

2.7.1 Backup set planning

Each backup set is tracked as a single object in Tivoli Storage Manager and has associated with it a *retention* value. As such, backup set is also termed as instant archive since it can also be used as an archive of client data retained for a defined period of time. The *retention* value only governs how long the Tivoli Storage Manager server retains the backup set in its volume history. If the retention period of the backup set expires, then those volumes will return to scratch status and the volumes becomes available for reuse. However, since each backup set is self-describing, the retention period has no significance to the backup set at all so long as the backup set volumes have not been reused. This would be the case if you have copied the backup set to a local file or to CD media. At any point of time, the backup set can be re-defined back to the same or different server using `define backupset` command.You can therefore use this feature to provide for your Tivoli Storage Manager client an instant archive and to keep the backup for as long as its required. A new retention period is assigned each time the backup set is defined.

2.7.2 Server/client media support

If you want to restore the backup set directly onto the client, without requiring to contact the Tivoli Storage Manager server, then the media which you use to create the portable backup set must be available and supported on both the server and client. There are a few ways to do this.

- Use a tape device which is available on both client and server. There are limitations on the exact devices supported for this, so you should consult the Tivoli Storage Manager Web pages for up to date information
- You could use a sequential device class written to disk and transfer the files to the client disk.
- Since many systems support a CD-ROM type device, you could either create the backup set on disk and use a CD writer program to copy the set onto CD, or if the operating system supports writing directly to a CD, JAZZ, or ZIP drive, you can use it directly in Tivoli Storage Manager by defining a REMOVABLEFILE device class. This type of class is also useful for transferring backup sets from one Tivoli Storage Manager server to another.

2.8 Restore

To restore a file, a directory, or even the whole machine, you need to know just two things: what you want to restore (file name, directory), and optionally, from when (point-in-time), if you want to restore a file other than the most recent one. You do not need to know where the data actually is. When you request a file, Tivoli Storage Manager finds out the location of that particular file version from its database.

To restore files, specify the directories or selected files, or select the files from a list or GUI window. By default, only ACTIVE file versions will be available for selection; however, INACTIVE versions can be easily specified. You can restore files to their original location or specify a different directory. Collision options control whether existing files of the same name are replaced or not.

Figure 35 shows a schematic of the restore operation.

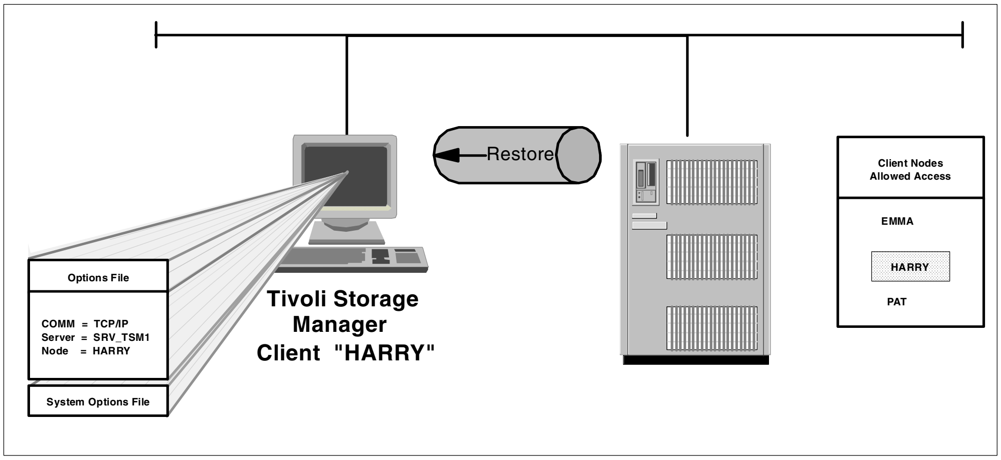
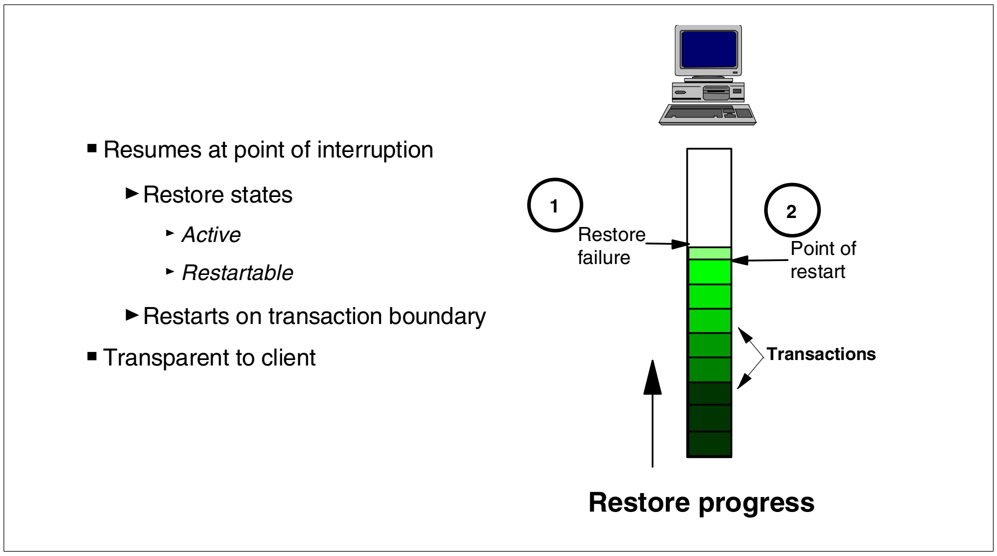


Figure 35. Restore in progress

You can restore using the command-line interface, GUI or Web browser interface (where available on the client operating system platform). The browser interface allows you to initiate the restore remotely so that you do not need to be physically located at the system being restored.

2.8.1 Restartable restore

When you are running a normal file restore operation, Tivoli Storage Manager keeps track of the files that you have already restored, so that it can retry the operation should you have any network problems. If the restore operation terminates prematurely for any reason, such as network failure, the session state remains in the database so that it can be restarted from the last completed transaction. This is known as a restarting the restore and saves you from having re-send files which have already been restored to the client before the session aborted. Figure 36 shows how a restartable restore is handled. There is a separate option in both the GUI and CLI for resuming the restore operation.



- Resumes at point of interruption
  - Restore states
    - *Active*
    - *Restartable*
  - Restarts on transaction boundary
- Transparent to client

Figure 36. Restartable restore processing

2.8.2 Point-in-time restore

A point-in-time operation restores the specified objects to the state that existed at a specific date/time. A point-in-time restore is supported on the filespace, directory, or file level. You must specify a sufficiently long retention period in the management class to enable this to occur. The general recommendation if you want to be able to provide a point-in-time restore capability for any time, say up to one month previously, you should set the parameters VEREXISTS and VERDELETED to NOLIMIT and RETEXTRA and RETONLY to at least that number of days. In this way, it does not matter how many times the files change in the restorable period, you will always have enough versions stored to be able to perform the restore.

Figure 37 shows a situation where data is being expired as new files are backed up. Tivoli Storage Manager can restore any file version from January 7th to January 3rd (but not January 2nd or January 1st since they are already expired because of the RETEXTRA parameter).

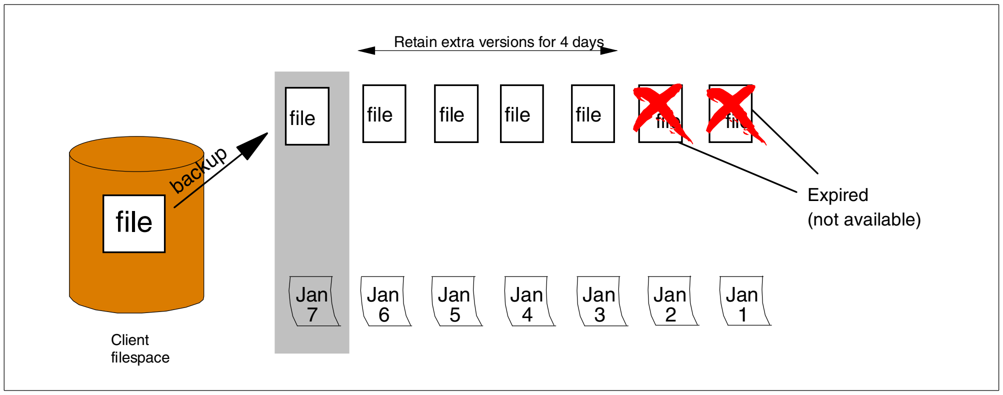


Figure 37. Expiration and point-in-time restore

Suppose that we have many backup versions stored, as shown in Figure 38. We had only fileA and fileB existing on our client at the previous backup. On DAY1, fileB is deleted which is picked up during the DAY1 backup. On DAY2 fileC was created. Sometime after this, fileD, fileX and fileY are also created and backed up. We also update fileA, creating a newer version and delete fileC. So, as at the present, DAY10, we have fileA, fileD, fileX and fileY on our workstation. We now request a point-in-time restore for DAY2 of this directory. Tivoli Storage Manager restores fileC, which had been deleted and replaces fileA with the older version (option replace=yes). Our machine now has fileA and fileC as at DAY2, and fileD, fileX, and fileY from the most recent position.

A point-in-time restore observes the following rules:

1. A file created after DAY2 is not restored; only the changes made before the date/time are restored (fileA in our example).
2. Files that were created on the client after the point-in-time date are not deleted (fileD, fileX, fileY in our example).
3. A point-in-time restore will restore files deleted after the point-in-time date (fileC), but not files deleted before (fileB).
4. Tivoli Storage Manager restores file versions from the most recent backup before the specified point-in-time date.

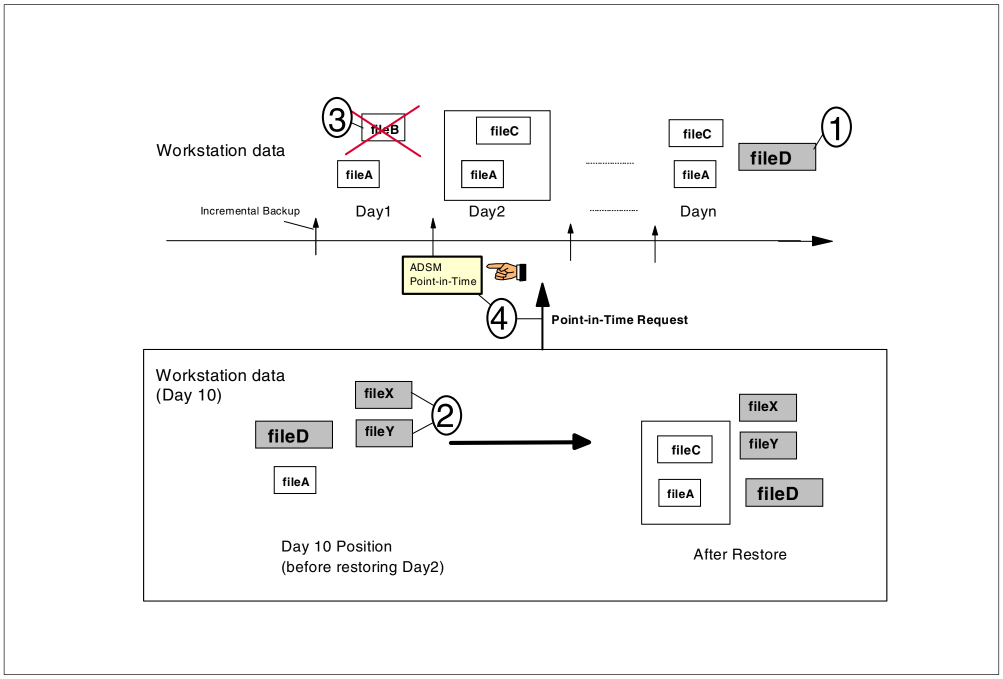


Figure 38. Point-in-time rules

You must be performing incremental file backups on a Tivoli Storage Manager Version 3 or later server in order to support a point-in-time restore. The Tivoli Storage Manager server is only notified when files are deleted from a client filespace or directory during an incremental backup. Selective and incremental-by-date backups do not notify the server about deleted files. Run incremental backups at a frequency consistent with possible restore requirements. For further information, see 2.4.9, “Backup special considerations” on page 45.

Figure 39 shows another scenario for point-in-time restore. If we try to restore a file from January 18th, 10:30, it can use Monday’s position to recover that file. On the other hand, if you need a file from January 22nd, the most recent image that Tivoli Storage Manager can use is the data from the previous day (January 21st) because we have not yet run a backup on January 22nd.

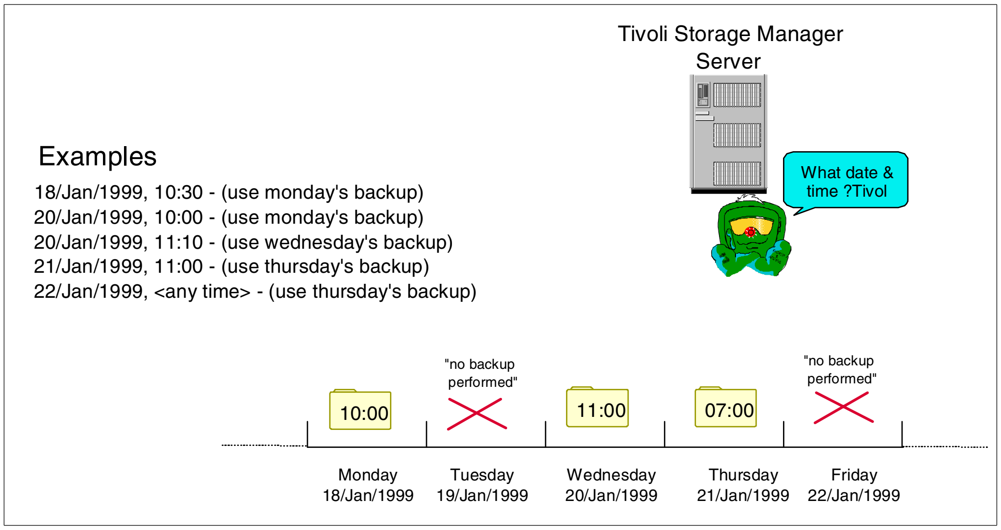

Tivoli Storage Manager Server
Examples
18/Jan/1999, 10:30 - (use monday's backup)
20/Jan/1999, 10:00 - (use monday's backup)
20/Jan/1999, 11:10 - (use wednesday's backup)
21/Jan/1999, 11:00 - (use thursday's backup)
22/Jan/1999, <any time> - (use thursday's backup)
What date & time ?Tivol
"no backup performed"
"no backup performed"
10:00
11:00
07:00
Monday 18/Jan/1999
Tuesday 19/Jan/1999
Wednesday 20/Jan/1999
Thursday 21/Jan/1999
Friday 22/Jan/1999

Figure 39. Point-in-time restore examples

2.8.3 Logical volume restore

Tivoli Storage Manager gives you the option to restore the entire logical volume from the full image backup, typically for a raw logical volume, or with options to restore and bring the logical volume status to the most current backup status (for a filesystem). Where there is a filesystem associated with the logical volume, if you have also been doing regular incremental backups of that filesystem, you can use the INCREMENTAL option in the *restore image* command. This will request changes to files recorded at the server since the full image backup was taken and will restore these also after the image restore, so that it is brought up to date. You can also specify the DELETE option with the *restore image* command, which will query the server for deleted files since the image was taken and remove these files. Note that the RESTORE IMAGE command will completely overwrite the existing logical volume and its filesystem.

2.8.4 Backup set restore

Restoration of a backup set can be done on a complete filespace level or by selecting individual files. You have the option to restore from backup set using either:

- Server based backup set restore from Tivoli Storage Manager server.
- LAN-free or local backup set restore directly via the Tivoli Storage Manager client from locally attached devices.

To restore from the backup set, the node name of the client must match the one which was defined in the backup set. If using LAN-free restore, the backup volumes are mounted by the backup-archive client on the client through normal operating system device drivers and file system media such as CDROM, JAZZ or ZIP and disk.

2.8.5 Cross-platform restore

Tivoli Storage Manager allows you to restore files from one client node to another client node. The restoring client node must be able to understand the filesystem of the original backup. For example, you could restore a FAT filesystem made from a Win98 client onto an Windows NT client, since Windows NT understands the FAT filesystem. However an NTFS backup could not be restored onto a UNIX system because it is not compatible. You can specify the source and destination for a restore operation and provide a different destination directory path. You are prompted for your password using this feature, but ownership is not changed. For example, suppose that you back up a UNIX file (/home/alemos/file1) from user alemos, and this user has an identification number 201 (UID=201). If you try to restore this file to another UNIX machine, then it will still be UID=201, even if the user (UID=201) does not exist, or if it is another one (for example UID 201=CLAUDIA). This means that whenever you restore a file from another platform to a machine with different authorization rules, you must check the permissions and also the ownership for those files in the new machine.

Figure 40 shows a machine (HARRY) being restored to another machine (DBFARIC). In our example, both machines are from the same operating system release and all users are exactly the same, and thus recovery is straightforward.

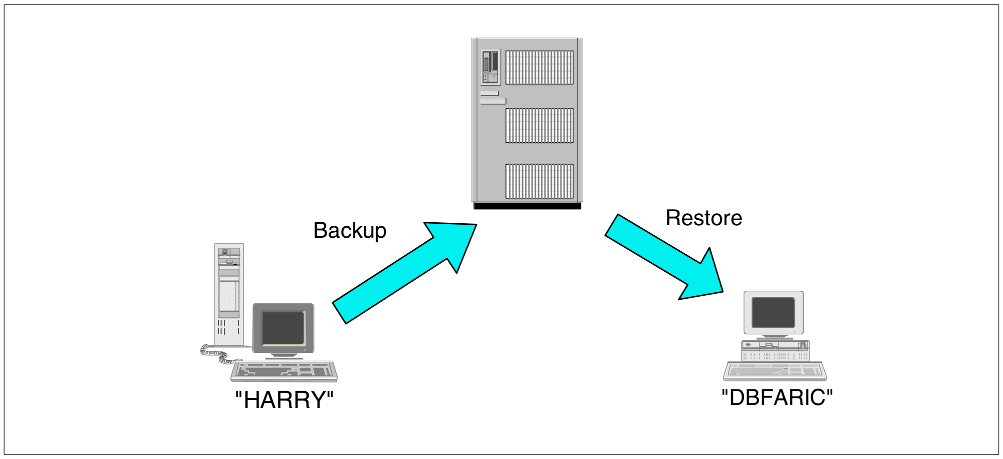


Figure 40. Cross-platform restore

2.9 Retrieve

The retrieve command obtains copies of archived files from the Tivoli Storage Manager server. You can specify either selected files or whole directories to retrieve archived files. The description option allows you to search for the descriptions assigned to the files when they were archived; you may decide to replace the files into the same directory from which they were archived, or in a different directory. Figure 41 shows a schematic view of the retrieve processing.

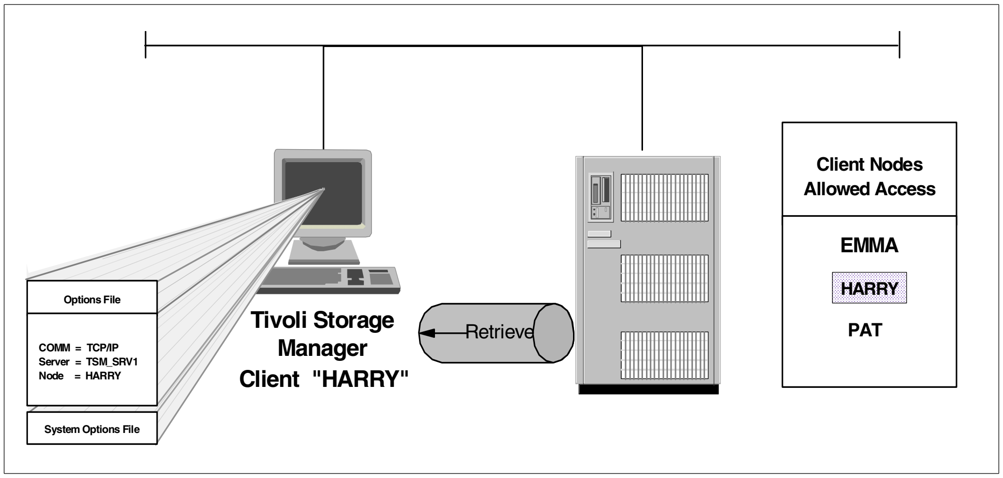


Figure 41. Retrieve in progress

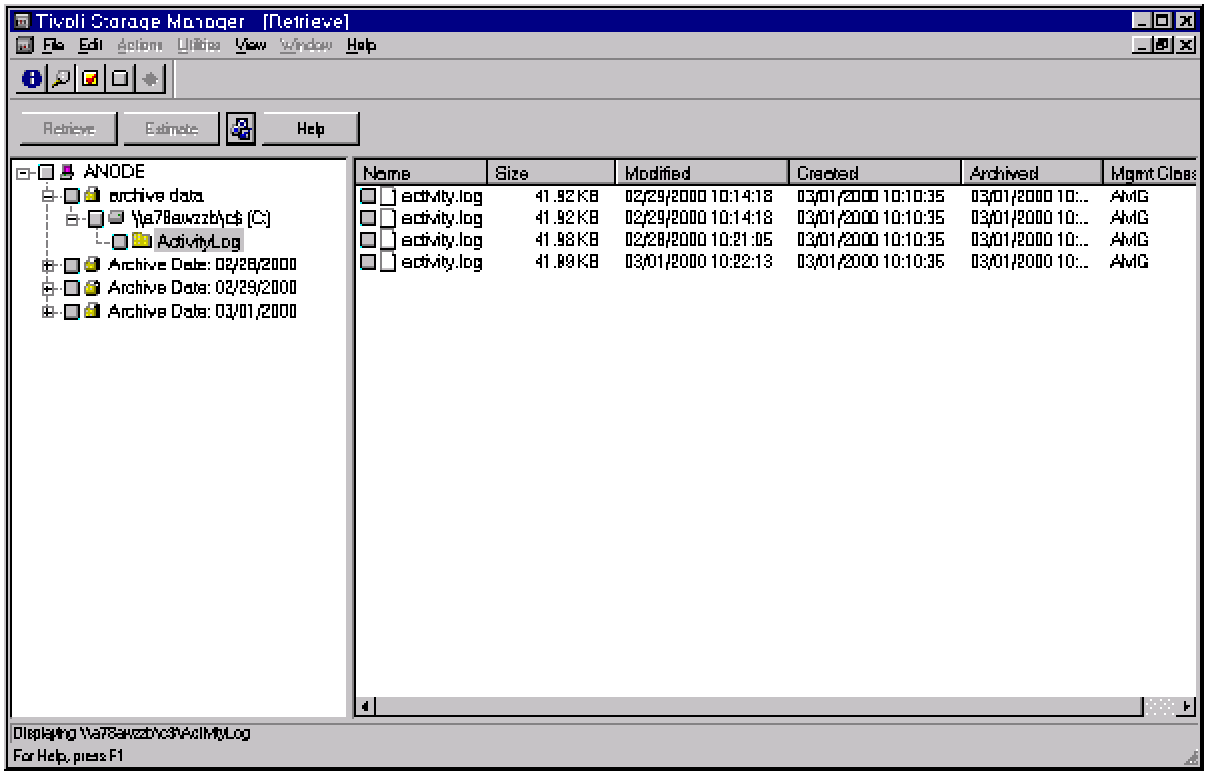
2.9.1 Retrieve key concepts

The retrieve option obtains copies of archived files from the Tivoli Storage Manager server. You can specify either selected files or whole directories to retrieve files. Use options, such as the description option, which allow you to search for descriptions assigned to the files when they were archived.

2.9.2 Packages

As explained in 2.5.1, “Packages” on page 46, you can search for a specific file within a package, or even retrieve the whole package. When you retrieve one or multiple files, Tivoli Storage Manager locates the volume where both directories and files are, so that you do not need to know which tape holds the data.

Figure 42 shows a retrieve window where you can examine all the previously archived packages. A search filter is also provided so that you only look for packages matching a certain string. You can see a series of entries of the same file with different weekly dates attached.



Name	Size	Modified	Created	Archived	Mgmt Class
activity.log	41.92 KB	02/29/2000 10:14:18	03/01/2000 10:10:35	03/01/2000 10:...	AMG
activity.log	41.92 KB	02/29/2000 10:14:18	03/01/2000 10:10:35	03/01/2000 10:...	AMG
activity.log	41.98 KB	02/28/2000 10:21:05	03/01/2000 10:10:35	03/01/2000 10:...	AMG
activity.log	41.89 KB	03/01/2000 10:22:13	03/01/2000 10:10:35	03/01/2000 10:...	AMG

Displaying \\a78awzzb\c$\ActivityLog
For Help, press F1

Figure 42. Retrieval from GUI showing ActivityLog package

2.10 Backup versus archive

Tivoli Storage Manager manages backup and archive objects differently, with respect to their versioning and retention. You use Tivoli Storage Manager backup/restore when you want to control the number of versions and retention period for files. Tivoli Storage Manager uses the management class definitions to enforce both the number of versions for a file (active and inactive) and the retention period. Because the incremental function rebinds the files to one management class, it is not possible to have different management classes for the same file (Tivoli Storage Manager incremental rebinds all versions). You can run backups to save changing files and use the management classes to have different controls for different files.

You use Tivoli Storage Manager archive/retrieve when you want to store a group of files for a period of time. Tivoli Storage Manager uses the management class definitions to enforce the retention period only. There is no way of specifying a version control for an archive file. The archive function does not use the bind/rebind concept. The only case when an archive file is managed differently is when you delete the archive management class that was controlling files. In this case, those files are controlled by grace period settings.

As a rule of thumb, you can think of backup as the process for storing all your vital and ever-changing data, which is normally performed by the daily backup process. All other long-term retention requirements (weekly, monthly, yearly) can be handled by the archive function. The portable client backup set (described in 2.7, “Backup set” on page 50) is an alternative to the archive function. The advantage is that you can create this set from the files already present in the Tivoli Storage Manager storage pools, so no re-sending of the client data is required. The disadvantage is that the backup set can be made only at the filespace level, so this granularity may be not suitable for your needs.

2.11 Other considerations

In this section we cover some other subjects which affect all of your client operations.

2.11.1 Scheduling

In our examples of the various client operations, we have shown how they operate from an end user perspective, that is, by using the different interfaces available. In a typical production environment, the backup and other operations which protect the client data should be scheduled, so that we can be sure they regularly execute and can see if or when something goes wrong. Tivoli Storage Manager provides you with a client scheduling interface, which interacts with the server Central Scheduler, for this purpose. Another option for scheduling is to use your own or third party scheduler to run scripts on your clients, comprising the appropriate client commands.

If you use Tivoli Storage Manager’s own central scheduling, the administrator defines appropriate schedules on the server to perform the Tivoli Storage Manager tasks automatically. Central scheduling is a cooperative effort between the server and each client node in that the client must also be running its own scheduler process so that the client and server can contact each other to correctly run the scheduled operation. The client scheduling process should normally be configured to start automatically each time the client boots, to avoid missing schedule execution and compromising data security. There are two methods used to control how the client and server make contact to run a schedule: *client polling* and *server prompted*. These options and scheduling in general, are discussed further in Chapter 4, “Scheduling” on page 73.

2.11.2 Compression

You have the option to specify that each client should compress its files or other objects before sending them to the Tivoli Storage Manager server. Compression is available for both backup and archive operations. Enabling client compression will decrease the network traffic between client and server (because you are sending a smaller quantity of data) at the expense of requiring more client CPU resources to perform the operation. Therefore, the decision to enable client compressions needs to made individually for each configuration. If you are using client compression, then the client will also automatically decompress any objects which are sent back to it from the server when the reverse restore or retrieve operation is requested. Objects which are compressed also ultimately take up less storage space in the Tivoli Storage Manager server storage pools, reducing resource requirements.

If you do not enable client compression, the files will be sent at their full size to the Tivoli Storage Manager server. It is likely that the sequential storage device, for example, a tape drive, can perform hardware compression. If this is the case, you will still get the benefit of reduced space required in the storage pool. Note that if a client has already compressed the files, then a compression-enabled tape drive will not normally be able to compress it further. Compression rates vary considerably depending on the type of data presented.

When using client compression, (*compression* set to *yes*) be aware that some files may actually grow when you try to compress them.This normally happens when the file has already been compressed by some other mechanism like ZIP or TAR and cannot be compressed further. Tivoli Storage Manager detects this condition and you can set options for the client which determine what should happen. You can either roll back the file and send it again without compressing or you can continue the compression operation to completion. Rolling back the file will cause a retry operation to be signalled and will probably increase your backup time if there are a lot of this type of file. On the other hand, forcing the compression operation when it ends up increasing the size of the file means you will be using more space in your storage pools. You should pick the option which best meets your file mix and environment.

2.11.3 Client authentication

A client is normally required to authenticate itself to the Tivoli Storage Manager server before being allowed to send or receive objects. The mechanism for this is a password, which is associated with each client when it is registered to the server. The password/authentication exchange guards against impersonation on either side by ensuring both that the client is a legitimate node and that the server is in fact the real server. The authentication mechanism does not transmit the actual password across the network so there is no risk of interception. The Tivoli Storage Manager administrator can disable authentication completely, which means that no password is required, however this is not normally recommended.

Assuming you are using authentication, the password access option can be set to prompt or generate. If it is set to prompt, which is the default, then the Tivoli Storage Manager server will ask you to supply a password each time you request backup, restore, archive, and retrieve services. If it is set to generate, then Tivoli Storage Manager automatically generates a new password for your client node each time it expires, encrypts and stores the password in a file on the client. The encrypted password will be automatically retrieved from the file whenever you request services. You are not prompted for the password. This option is particularly useful for scheduled operations. As we have discussed in 2.11.1, “Scheduling” on page 60, we want to have the scheduler client started automatically whenever the client is booted. Setting the PASSWORDACCESS option to *generate* avoids having to manually supply the password or include it in startup scripts. When using the Tivoli Storage Manager Web Backup-Archive client, you also need to set this option to *generate*.

Chapter 3. Data storage policy

A data storage environment consists of three types of resources: machines, rules, and data. The machines are computers which contain data that must be backed up. The rules specify how the backed up data is to be treated. Basically, a data storage policy defines the relationships between these three resources as illustrated in Figure 43.

Tivoli Storage Manager has entities that group and organize the resources and define relationships between them. A machine, or node in Tivoli Storage Manager terminology, is grouped together with other nodes into a policy domain. The domain links the nodes to a policy set which consists of management classes. A management class contains rules called copy groups that it links to the data.

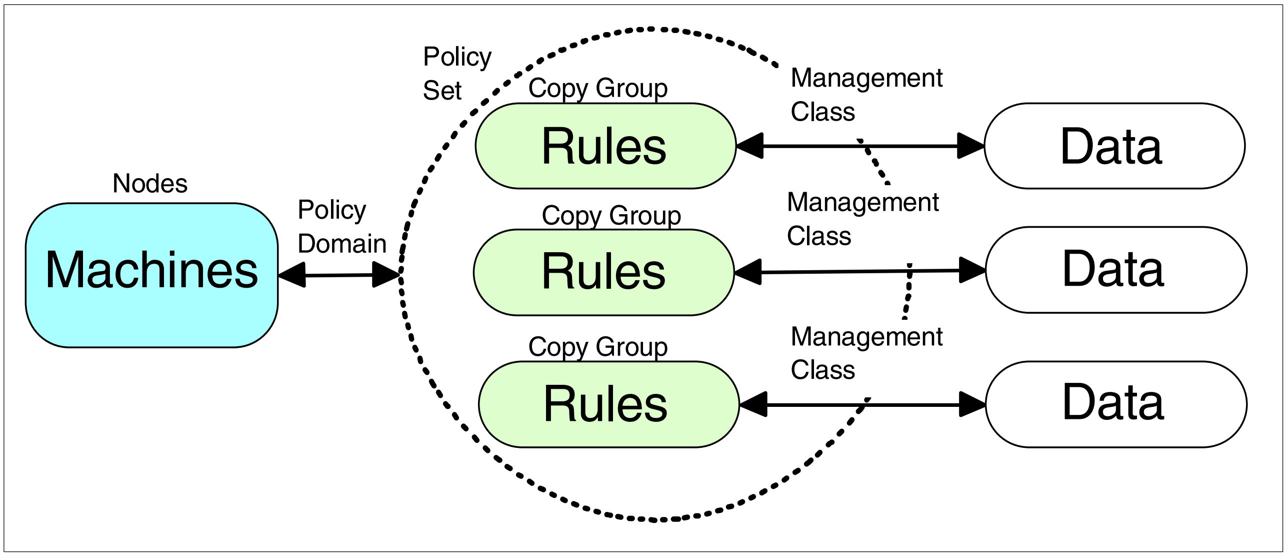


Figure 43. Data storage policy relationships and resources

3.1 Data storage policy components

The hierarchical structure of the Tivoli Storage Manager policy components is represented graphically in Figure 44. If we were discussing the technical steps required to configure a policy domain, it would be natural for us to start at the top and work our way down, since the policy domain has to exist before the policy set and so on. Examining the diagram leads to the realization that most of the data storage policy components exist solely to provide flexibility in our configuration or to serve as containers for rules. For the purpose of understanding the concepts behind the Tivoli Storage Manager hierarchy, we will begin our discussion with copy groups and work outwards toward the policy domain.

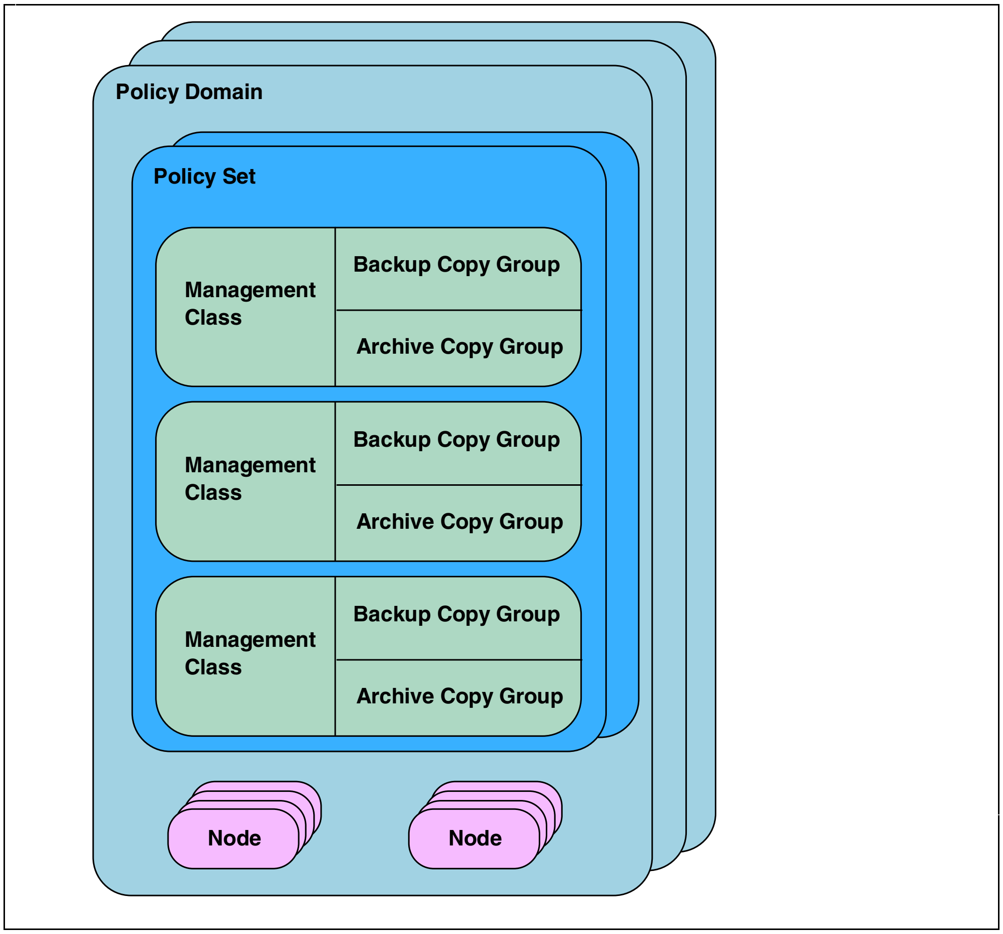


Figure 44. Data storage policy components

3.2 Copy groups

The copy groups consist of rules used to govern the retention of data. There are two types of copy groups: a backup copy group holds the rules for backup data and an archive copy group holds the rules for archive data.

While these two copy groups serve different purposes, they also share some common ground. They both specify where to store the data sent to them from backup/archive operations. The copy group destination parameter specifies a valid primary storage pool to hold the backup or archive data. The copy group bridges the gap between data files and storage pools as illustrated in Figure 45. It shows different types of data flowing through the copy groups and into the storage pools. Note that there is not necessarily a one-to-one relationship between copy groups and storage pools. It is possible to have just one storage pool which is the destination for all the copy groups.

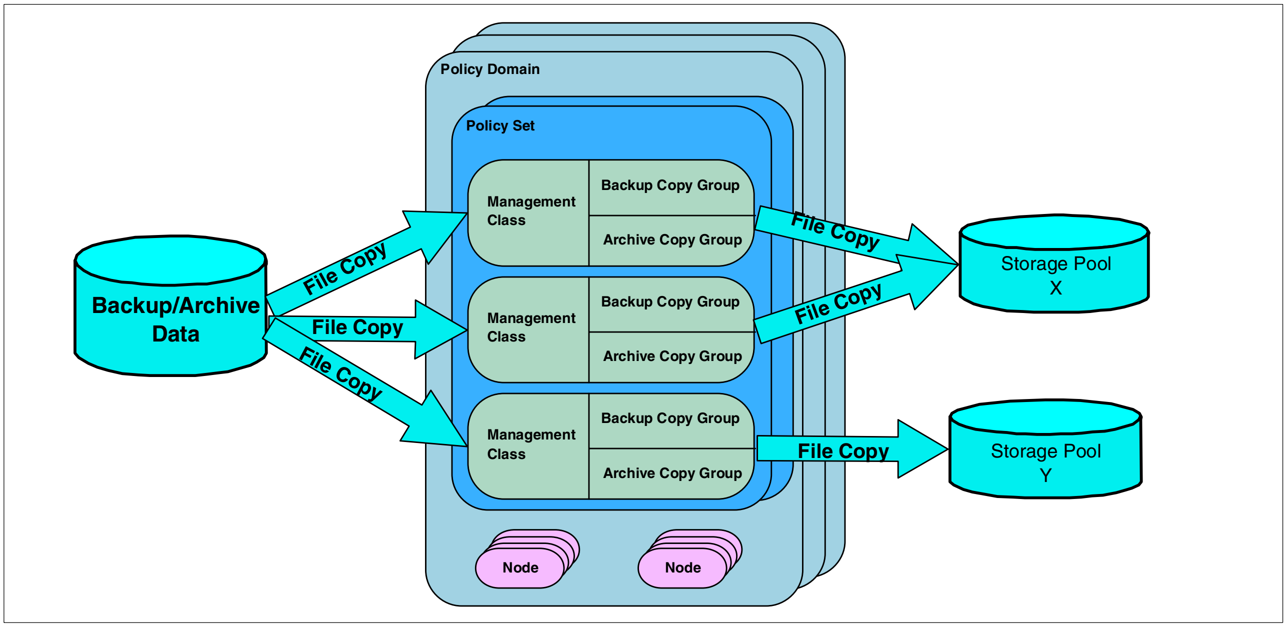


Figure 45. Data flow through copy groups

Both copy groups also need to know what to do with files that are modified during a backup or archive operation. When we say that a file is modified during backup, we mean that it is modified after Tivoli Storage Manager examined it for its details but before it was completely backed up to the server. This sort of backup is referred to as a "dirty backup" because the file is in an inconsistent state and may not restore properly. The copy serialization parameter provides four options to deal with this problem:

- The *shrstatic* setting specifies that a file will not be backed up if it is modified during backup, but multiple attempts will be made to back up the file. If the file continues to be modified through each of these attempts, the file will not be backed up. The number of attempts can be controlled using the changingretries option in the client options file.
- The *static* setting specifies that a file will not be backed up if it is modified during backup and no additional attempts will be made.
- The *shrdynamic* setting specifies that a file will be backed up if it is modified during backup but multiple attempts will be made to back it up without modification first. If that cannot be done, then the file will be backed up anyway.
- The *dynamic* setting specifies that a file will be backed up even if it is modified during backup. There is no preliminary attempts to back up the file unmodified; it is backed up on the first attempt.

Note that the copy serialization works differently when doing an image backup. For an explanation, see 2.6, “Logical volume backup” on page 48.

3.2.1 Backup copy group

The backup copy group is concerned with two logical objects: the file and the file copy. A file is the actual data on a client node, while a file copy is a point-in-time copy of the file stored on the server. Another way to think of it is that the Tivoli Storage Manager server contains file copies and nodes contain files.

A file can be in one of two possible states: existing or deleted. When we talk about an existing file on a node, we mean a file that has been previously backed up and still exists on the node. A deleted file is a file that has been previously backed up and has been subsequently deleted from the node. This simple concept is important when discussing data storage rules.

A file copy can be in one of three states: active, inactive, or expired. An active file copy is the most current copy of the file; an inactive file copy is a previous copy or version of the file, and an expired file copy is a copy to be removed from the Tivoli Storage Manager server. A backup file copy is set to the expired state when it no longer conforms to the rules specified in the backup copy group.

Whether the file exists or is deleted, the file copy will always pass through the same states in the same order. A file copy will start out as active since it will be the first copy of the file and therefore the most current. Once the file changes and we take another file copy, the first file copy will change to inactive because we have a more recent one. Eventually, the first file copy will be expired based on one of two limits placed on it by our rules: number of copies or retention period.

The number of copies that we set in our rules specifies the total number of file copies to maintain in the Tivoli Storage Manager database. It is important to note that the specified number includes the active file copy. Thus, when we set the number of file copies to three, we are keeping one active copy and two inactive copies. When the number of copies is exceeded, the oldest copies are removed from the database.

The retention periods that we set in our rules will specify the length of time that we will retain *inactive* file copies. It is important to note that there is no retention period for active file copies; they exist as long as the file exists on the node.

Whether or not the file exists on the node will affect which rules are used to expire the file copies. If the file exists, the following two backup copy group parameters are in effect:

- verexists — specifies the number of file copies, or versions, to keep. This number includes active and inactive file copies.
- retextra — specifies how long to keep inactive file copies. When a file changes from active to inactive, it will be kept for retextra days and then removed. It is important to note that the retention period starts from when the file copy becomes inactive not from its original backup date.

If the file has been deleted, the active file copy will be made inactive. At this point, there are only inactive file copies for this data in the Tivoli Storage Manager server and the following parameters apply:

- verdeleted — specifies the number of file copies to keep after the file has been deleted.
- retonly — specifies how long to maintain the last file copy of the data. This is the number of days to keep the last copy only and does not apply to other inactive file copies.

See 2.4.1, “Backup key concepts” on page 37 for examples illustrating these concepts.

3.2.2 Backup mode and frequency

The backup copy group defines two other attributes that control the way that backup data is handled: *mode* and *frequency*. The *mode* parameter specifies how files are to be selected for incremental backup. Setting the mode to modified will allow a file to be backed up only if it has changed since the last backup. The absolute setting will allow files to be backed up, regardless of whether they have changed or not. The latter value would only be used for special cases; the default value is modified.

The *frequency* parameter specifies how often to allow a file to be incrementally backed up. This parameter only affects client-selected *incremental* backups; it does not apply to *selective* backups, since this backs up data regardless of whether it has changed or not. For a file to be backed up during an incremental operation from a node, it has to satisfy three conditions:

1. Domain and include-exclude statements allow the file to be considered for backup.
2. The file satisfies the mode setting. That is, if the mode is set to modified, the file must have changed to qualify for backup. If the mode is set to absolute, then the file is automatically allowed to be backed up.
3. Difference between the server time and the active file copy timestamp must be greater than frequency setting. The frequency is converted to hours to compare to the timestamp difference.

For example, consider a file called /home/admin/redbook.doc that is eligible for backup in the include-exclude list and that has changed since the last backup at 8 a.m. this morning. The server time is 11 a.m. when an incremental backup is started, so the difference between server time and the file copy time is 3 hours. If the frequency is set to 1 day, then 24 hours must pass between incremental backups before a file is backed up again. Therefore, the file called /home/admin/redbook.doc will not be backed up, since 3 hours is less than 24 hours. The default setting for this parameter is 0 which means there is no minimum required elapsed time between backups.

3.2.3 Archive copy group

The archive copy group works with entire archives as single unique entities so it has fewer rules. Since there is only ever one copy of a particular archive, we do not have to worry about rules to manage versioning. We still have to specify the retention period for the archive object and that is done with the *retver s*etting. It specifies the number of days to retain the archive copy from the day of the archive operation.

There are three other parameters that the archive copy group uses to handle archive data: *mode*, *frequency* and *copyserialization*. The mode parameter has been discussed, but the archive copy group only allows the value to be set to absolute. This makes sense, since the archive copy group does not link previous archives to the current archive and therefore cannot determine if anything has been modified. The archive copy group has a frequency parameter, but it can only be set to the value CMD which means the archive can be performed on demand.

Copyserialization operates in the same way as for backup and has already been discussed in 3.2, “Copy groups” on page 64.

3.3 Management class and explicit binding

The management class serves two purposes: it contains copy groups and associates data to them. A management class must contain at least one copy group. This copy group could be for backup, or archive, or the management class could contain both a backup and an archive copy group. Figure 46 shows the basic structure of a management class with both copy groups defined. It also illustrates how a management class links the backup/archive data to the rules defined in a copy group. The link is very granular and can be assigned to a single file or groups of files. When a file is linked to a management class, it is said to be bound to the management class.

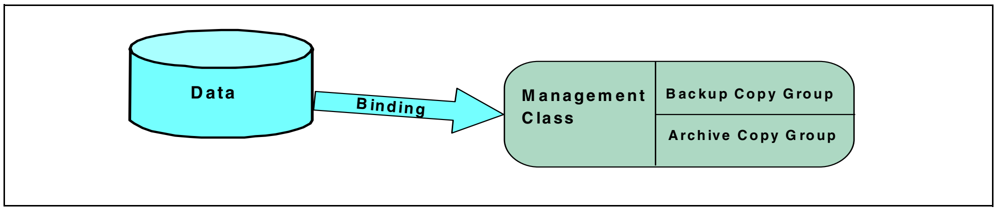


Figure 46. Binding data to the management class structure

There is a special instance of a management class called the *default* management class. There is only one for each logical grouping of nodes (policy domain), and it contains the rules that you want used for your data unless you explicitly bind it to another management class. Therefore, there are two ways of binding data to a management class: default and explicit. Unless an object is explicitly defined, the default management class is used.

Explicit binding to a management class can be accomplished using three different methods: one for backup data, one for archive data, and one for directory data. While you are not required to use any of these methods, they are very powerful and important tools for effective data storage management.

Binding your backup data to different management classes allows you to manage different types of files with different sets of rules. Backup data is bound to a management class using the include option of the Tivoli Storage Manager client’s include-exclude list. Figure 47 shows an example of an include statement that assigns the backup file /home/admin/redbook.script to a management class called redbook while allowing the rest of the backup files in /home/admin to go to the default management class. The binding is actually done during the backup operation.

```
include /home/admin/.../*
include /home/admin/redbook.script redbook
```

Figure 47. Include option example

Image backups are also bound using the include/exclude list, however, the `include.image directive` is used. This example in Figure 48 shows how to bind all image backups to the imageMC management class.

```
include.image /.../* imageMC
```

Figure 48. Image management class example

Archive data is bound to a management class using the `archmc` command line option as illustrated in Figure 49. The file /home/admin/redbook.doc will be bound to the management class redbookarchive. Without the archmc option, this data would have been bound to the default management class.

```
dsmc archive -archmc=redbookarchive /home/admin/redbook.doc
```

Figure 49. Archive management class example

Directory data can be bound to a management class using the `dirmc` client option as illustrated in Figure 50. All of the directories from this client will be bound to the directory management class. Without this option, the directory information would go to the management class with the longest retention period. You can use this option to speed up your restores by binding your directory data to a management class which uses only a primary disk storage pool instead of tape. Directory information does not take up very much space so only a small storage pool is required. The directory information is the first thing that is rebuilt during a restore, and by making it rapidly accessible, your restore time will be improved.

```
dirmc directory
```

Figure 50. Directory management class example

3.4 Policy set

A policy set is a group of management classes. There can be multiple policy sets within a policy domain, but only one of them can be active at a time. That is to say, the active policy set contains the only group of management classes that can be bound to the data within the domain. Management classes in other policy sets are not available unless you activate the policy set that contains them.

A policy set can contain many management classes but only one of them can be the default. The basic structure of a policy set is shown in Figure 51 and illustrates that the policy set is used primarily for flexibility. It allows us to group management classes and assign one of them as a default for the policy domain.

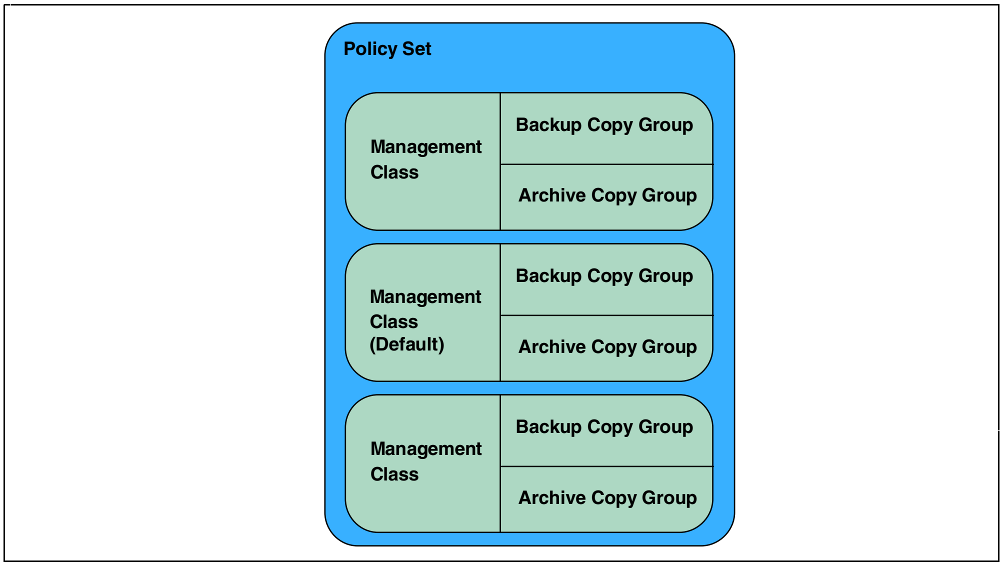


Figure 51. Policy set structure

The active policy set is a special entity in the policy domain and cannot be changed directly. To change it, you must define your rules in a policy set which is subsequently validated and activated. The activation process takes a snapshot of the policy set and places it in the active policy set. It is important to note that the active policy set is a point-in-time snapshot of the originating one. Therefore, further changes to the originating policy set will have no effect on the active policy set until the changed one is validated and activated.

The validation process checks that your policy set is complete and valid. It will check the management classes and copy groups and make sure that the policy set has a default management class. It will also make sure that the copy groups point to valid primary storage pools.

The activation process will verify that the default management class and copy group definitions are correct (using the same checks as the validation process) before activating the policy set. Activating the policy set means that it is copied to the active one.

3.5 Policy domain

A policy domain is a way to group Tivoli Storage Manager clients depending on how you want to treat their data. It also contains policy sets that have all of the rules that you want to apply to your data. Figure 52 shows a simplified view of a policy domain with multiple policy sets and nodes contained within it. As discussed, there can be many policy sets but only one is active.

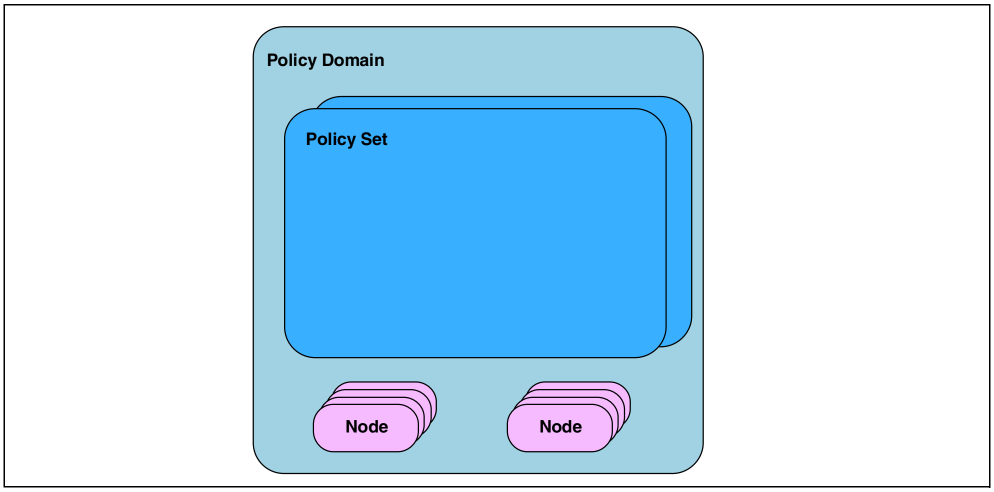


Figure 52. Policy domain structure

A policy domain allows you to logically group the machines in your organization according to:

- Default policy—the default set of rules to apply to the clients. The rules define the storage management policy including how many copies of data to keep and how long to keep them. The default management class of the active policy set contains the default rules applied to the domain.
- Administrative control—access to the client data and the policy rules can be restricted to certain administrators by allowing or disallowing the administrators access to the policy domain.

Let us consider a typical organization consisting of UNIX and Windows NT machines. The UNIX machines are large file servers and need many copies of their data maintained for a long period of time. The UNIX support group is the only group authorized to access the UNIX machines. The UNIX policy domain would hold all of the UNIX machines, and would only be accessible to the UNIX Tivoli Storage Manager administrators. The default policy in this domain would apply only to the UNIX machines.

The Windows NT machines are workstations and need a few copies of their data maintained for a short period of time. The Windows NT support group is the only group authorized to access the Windows NT machines. The Windows NT policy domain would hold all of the Windows NT machines, and access to it would be restricted to the Windows NT Tivoli Storage Manager administrators. The default policy in this domain would only apply to the Windows NT machines.

This is a good example of how default policy and administrative control can be used to break up your organization into policy domains.

3.6 Policy management

The policy domain and all of its subordinate entities contain your data storage definitions but it does not actually enforce them. The copy group rules are used during the client backup process to mark inactive file copies as expired. They can be expired based on the number of copies or retention period. Then a server process, called inventory expiration, is executed to actually remove the database references for the file copies. This process will remove backup and archive data entries in the database. It is important to note that the actual data is not removed from the storage pools; only the pointer in the database is deleted.

The inventory expiration process can be run manually, automatically or by schedule. By default, a Tivoli Storage Manager server will run this process daily but that can be controlled using the EXPINTERVAL parameter in the server options file which specifies the number of hours between automatic expiration processing.

Since the inventory expiration process can be quite CPU intensive, the preferred method for running it is to use the Tivoli Storage Manager scheduler to define this operation to run at a pre-determined, convenient time for your environment. The next chapter, Chapter 4, “Scheduling” on page 73, discusses how to use the Tivoli Storage Manager scheduler.

Chapter 4. Scheduling

Tivoli Storage Manager includes a central scheduling component that allows the automatic processing of administrative commands and client operations during a specific time period when the schedule is activated. An administrator is responsible for creating and maintaining the schedules in each policy domain.

Tivoli Storage Manager scheduling is split into two categories: administrative and client scheduling. The two categories differ in two key areas:

- Execution location—An administrative schedule performs some action on the Tivoli Storage Manager server while the client schedule can only execute on the Tivoli Storage Manager client.
- Domain privilege—An administrative schedule can only be managed by an administrator with system privilege, while the client schedule can be managed by an administrator with policy privileges in the client's domain. The granularity provided by this feature can be very useful when distributing management control across a large enterprise.

For both types of schedules, there are four key pieces of information:

- A command that needs to be executed
- When the command executes
- How long the command takes to complete
- How often the command needs to be repeated

The command that you run may be an incremental backup (client schedule) or a storage pool migration (administrative schedule) which should be run every day at a particular time. You will also have to estimate how long the command will run so that you can synchronize your schedules and balance load on the server.

All schedule start times are based on the Tivoli Storage Manager server clock, regardless of the time zones where the clients are located.

4.1 Administrative schedules

An administrative schedule is a directive to trigger some sort of action on the Tivoli Storage Manager server. It consists of a command or sequence of commands and some details on when the actions should happen. Any actions that are used on a regular basis to manage the Tivoli Storage Management environment should be defined as an administrative schedule. Administrative commands can be scheduled for use in tuning operations and to start functions that require significant server or system resources. Automating these operations to occur in a quiet period, say overnight, allows the administrator to ensure that server resources are available when needed by clients.

4.1.1 Scheduling concepts

The example in Figure 53 shows a series of operations that could occur in a Tivoli Storage Management environment and the sequence for those operations. The circle represents a clock, and the tick marks indicate the hours of the day. The actual start time and duration of the various operations are subject to your scheduling requirements, and the sequence should be carefully considered.

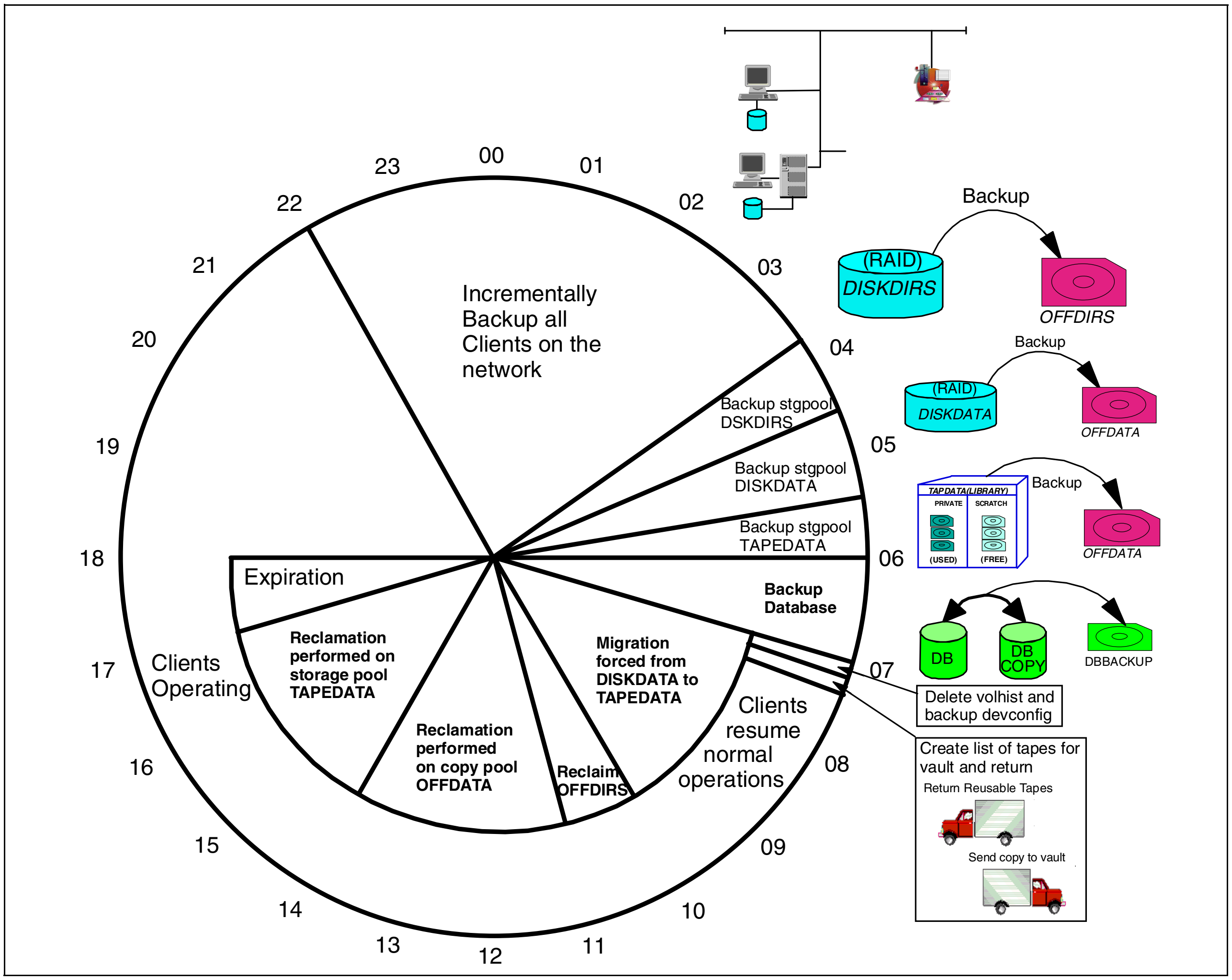


Figure 53. Client schedules

4.2 Client schedules

A client schedule is a directive to trigger an action on one or more Tivoli Storage Manager client machines. It is different from an administrative schedule in that it specifies that an action be performed on the Tivoli Storage Manager client. The client scheduling system consists of a server portion and a client portion. The server part is integrated into the Tivoli Storage Manager process and is responsible for defining the schedule parameters and associating nodes with the schedule. The client scheduler is a separate process on theTivoli Storage Manager client and provides communication between the server and client. A client must be running its scheduler process to be able to execute scheduled operations. Otherwise, the operation will fail and will be logged as such in the Tivoli Storage Manager server.

The success or failure of each scheduled operation is recorded in the server event log, so the administrator can query to find out which, if any, failed. The client also keeps a local record of scheduled operations.

The definition of the client schedule includes the actions which are to be performed. This can include a single native Tivoli Storage Manager backup, restore, archive or retrieve command; a macro, which is a collection of native commands; or a command script. A scheduled command script is simply any script which could be executed within the client operating system, for example, a Windows batch command file or a UNIX shell or perl script. The command script could itself contain Tivoli Storage Manager client commands, plus logging, setup, error detection, post-backup procedures and so on, or might be used to schedule functions totally unrelated to Tivoli Storage Manager functions if required. It is therefore a very flexible general purpose scheduler.

A schedule is defined within the policy domain, so that only the client nodes contained within that domain are eligible to execute that schedule. After defining the schedule, the administrator then needs to specify which client nodes (from those in the domain) should execute the schedule. This action is called *associating* clients with a schedule. The administrator may choose to associate all the nodes in the domain, or just a subset, according to requirements. The associations can be changed at any time.

There are two scheduling modes available for establishing communication between the client and server to start a scheduled event, as shown in Figure 54. The selection of which mode to use is set in the client options file and is also dependent on the basic communication protocol used between the client and server. If the communication method is anything other than TCP/IP, then only the client polling method may be used. If TCP/IP is used, then the client may select either method. Optionally, the server can override the client's preference if required.

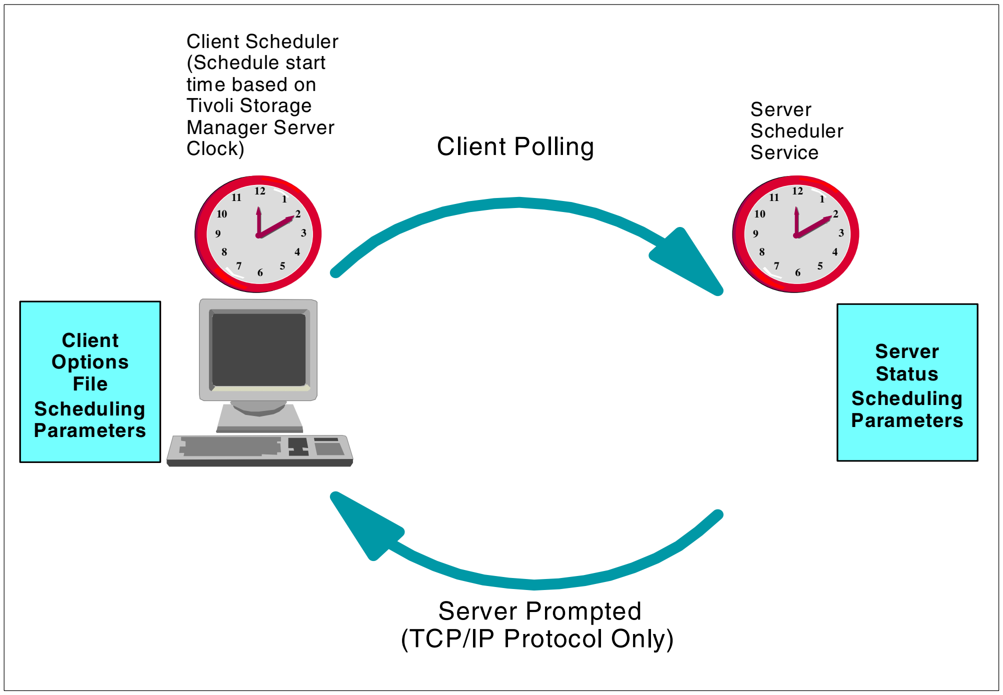

Client Scheduler (Schedule start time based on Tivoli Storage Manager Server Clock)
Client Polling
Server Scheduler Service
Client Options File Scheduling Parameters
Server Status Scheduling Parameters
Server Prompted (TCP/IP Protocol Only)

Figure 54. Client schedule types

4.2.1 Client polling

Client polling means that the client contacts the server to find out if there is a schedule defined for him and when he should run it. So that the client can respond dynamically to any amendments made to its schedules by the administrator, the client continues to contact the server at regular intervals, set in the options file. When a contact is made, if there is a schedule to execute, the client receives a start time from the server clock. The client then counts down to the start time, when it will perform the actions defined by the schedule. If there is no scheduled action to run between the time of contact and the next contact time, the client returns to sleep.

For example, a daily schedule may back up all of the client's file systems incrementally. Figure 55 shows a client polling schedule configuration in which the client keeps querying the Tivoli Storage Manager server from time to time about the next operation to run. In this case, the Tivoli Storage Manager server informs the client when the scheduler must run, but if it is yet not the right server time, the client waits for the right moment.

This mode has the advantage that you can have the server automatically assign randomized start times for each client's schedule execution, within a proportion of the schedule's window. This is useful if there are many clients which have to execute the same scheduled operation and you do not want to overload the server by starting them all simultaneously. This randomization function is not available when the schedule mode is server prompted.

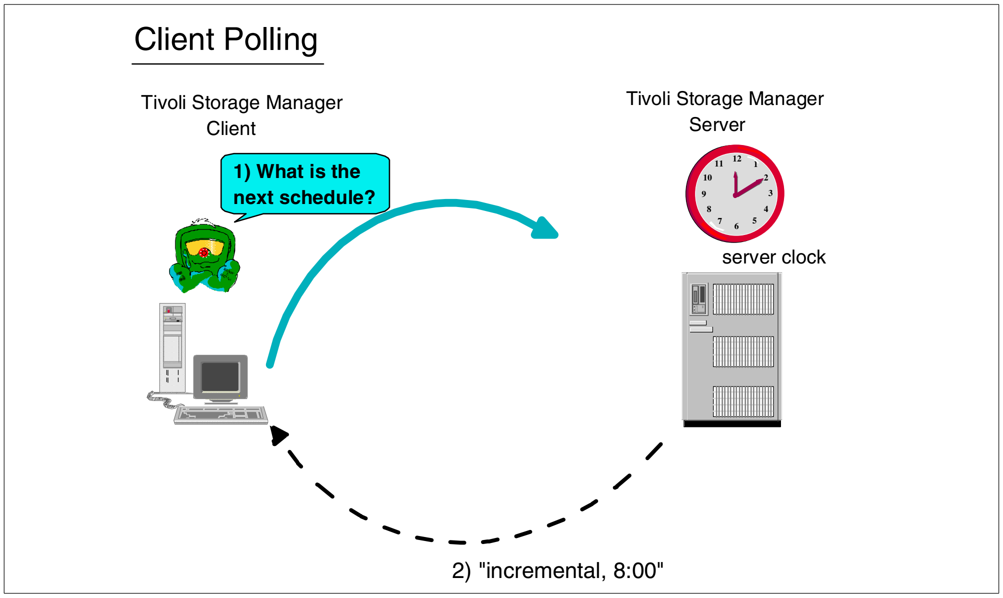

Client Polling
Tivoli Storage Manager Client
1) What is the next schedule?
Tivoli Storage Manager Server
server clock
2) "incremental, 8:00"

Figure 55. Client polling scheduling

4.2.2 Server prompted

In server prompted mode, the client first contacts the server to notify that it is running and is able to receive new schedule notifications. The client then sleeps until the server contacts it. When the server detects that a new schedule is ready to run, it contacts the client and informs that a new scheduling operation must be started. The client gets all the schedule definitions and starts the operation.

Figure 56 shows an example of a scheduled operation that the client must run.

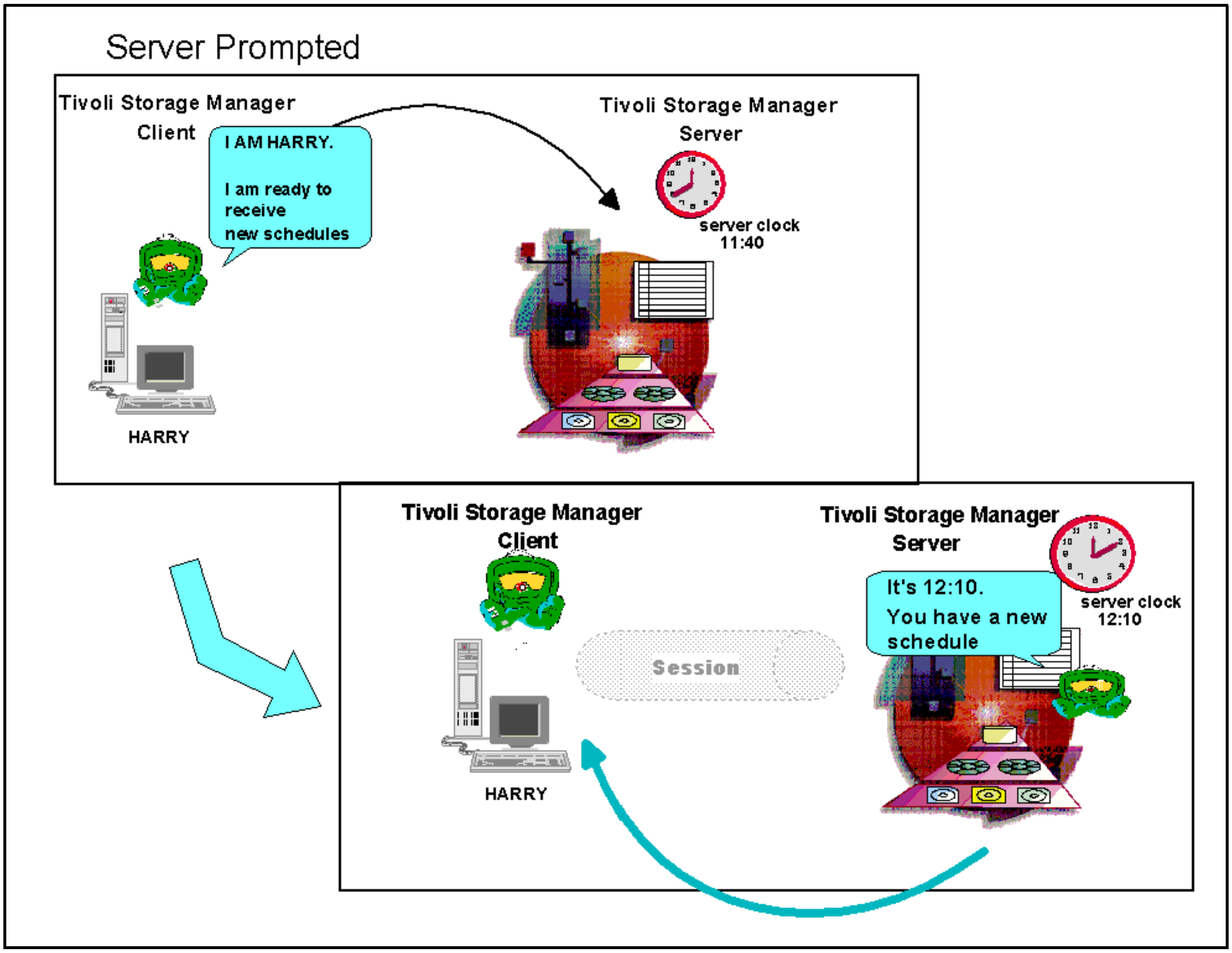

Server Prompted
Tivoli Storage Manager Client
I AM HARRY. I am ready to receive new schedules
Tivoli Storage Manager Server
server clock 11:40
HARRY
Tivoli Storage Manager Client
Tivoli Storage Manager Server
It's 12:10. You have a new schedule
server clock 12:10
Session
HARRY

Figure 56. Server prompted scheduling

If you use server prompted scheduling, you can quickly and easily rerun a failed backup process, because you can restart it from the server. The other advantage of server prompted scheduling is that you do not have the continued regular polling traffic from each client which is required by the client polling method.

4.2.3 One-time client schedule

As well as regularly scheduled operations, you may also define a schedule for one-time-only processing, on a single client or set of clients. This is called a CLIENTACTION schedule.

Unlike the regular client schedule, the definition of a one-time client schedule and association of the nodes with it, is done as a single command by the administrator. Once the schedule is defined, the client nodes associated with the schedule which are using server prompted mode, will receive the schedule and process the command virtually immediately. Clients using the client polling mode will receive the command the next time they poll the server.

The length of time the CLIENTACTION schedule exists on the server is defined by another administrator-defined parameter. After this time, the database will automatically remove it whether associated nodes have run the schedule or not.

4.3 Frequency and duration

The administrator defines the time period (that is, the *duration*) during which a schedule can start, and how often to repeat the schedule (the *frequency*). An example of a typical schedule frequency could be: every night at midnight. The start time is 00:00 and the period between startup windows is 24 hours. The duration is set to 2 hours. See Figure 57.

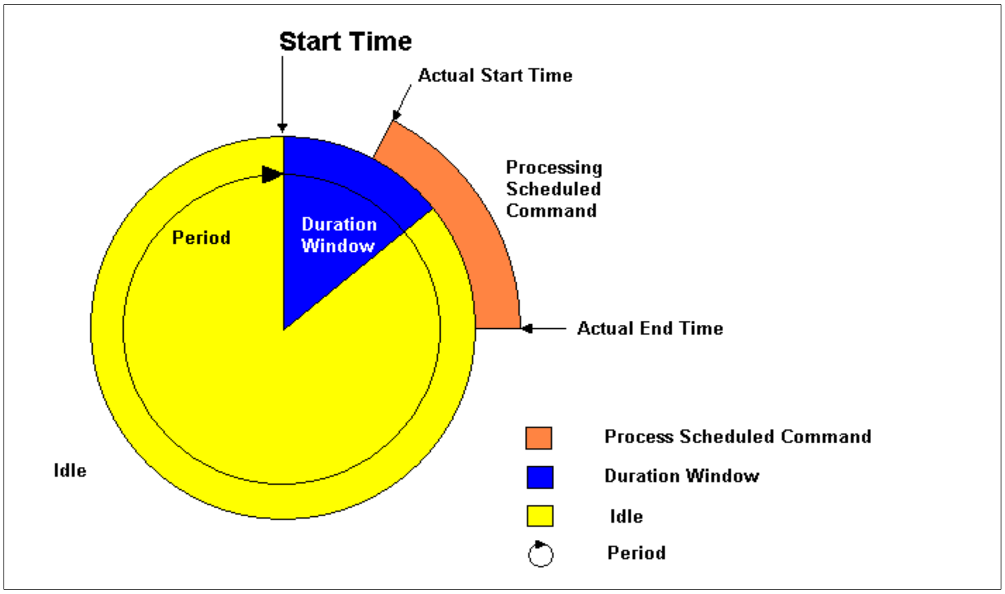


Figure 57. Schedule frequency

The actual time that a scheduled operation will start can be at any time during the duration window. If it is unable to start in this window, then it will be missed and recorded as such. After the period parameter between schedules has passed, the schedule will attempt to start again. The duration window therefore allows handling of a situation where the client is not running its scheduler process when the schedule is supposed to start, for example, if it has been powered off. So long as the client is available at some time before the end of the duration window, the schedule will be able to start.

The actual time that a scheduled event will end is dependent only on how long it takes the scheduled process to complete. It has no relation to the duration window. This is shown in Figure 57.

4.4 Retry and randomization

You can specify the maximum number of concurrent clients allowed to log on to Tivoli Storage Manager, as well as the maximum number of scheduled sessions allowed. If you restrict the number of scheduled sessions allowed on the server, a client is prevented from running a schedule when the maximum number of sessions has been reached. Through options which can be set globally at the server or individually for each client, the client can retry a certain number of times to run the schedule and with specified time interval between retries.

With the retry and randomization options, you have considerable flexibility to balance the network load.

Let us consider the scenario where you, as an administrator, associate 100 workstations with a backup schedule that has a window of 2 a.m. through 6 a.m. every Friday. You can use the randomization option to tell Tivoli Storage Manager to stagger the start times of 100 backup sessions so that they start at different times. In this way, you prevent a big bottleneck, which would occur if the 100 schedules started at 2 a.m., all at once!

4.5 Event log

Scheduled operations are recorded centrally in the event log at the server.

You can view which schedules ran successfully, were missed, and are scheduled to run in the future. You can create an exception reporting list. In this way you can view only those schedules that failed.

Detailed reports of the schedule are logged at client level. High level results are sent to the server to an event log, so you can view whether or not schedules ran successfully.

You determine how long a record is retained in the event log.

Chapter 5. Data storage

Tivoli Storage Manager represents data storage with administrator-defined objects: storage pools and storage pool volumes.

In the following sections we discuss how Tivoli Storage Manager automatically manages these pools and how you can define and check that the command used has performed the changes you require.

We also discuss some Tivoli Storage Manager concepts related to storage pools: how to use fewer tapes (reclamation), and how to reduce a client’s restore time (collocation). We provide a brief outline of protecting disk storage by using some form of redundant array of independent disks (RAID).

5.1 Storage pools

A storage pool is a collection of storage pool volumes; each storage pool represents one type of media. See Figure 58. For example, a storage pool for a 4 mm digital audio tape device (DAT) represents collections of only 4 mm tapes, and a storage pool for an IBM 3590 represents collections of only 3590 tapes. A storage pool created on a disk has files formatted under Tivoli Storage Manager as volumes and are collectively grouped in the storage pool.

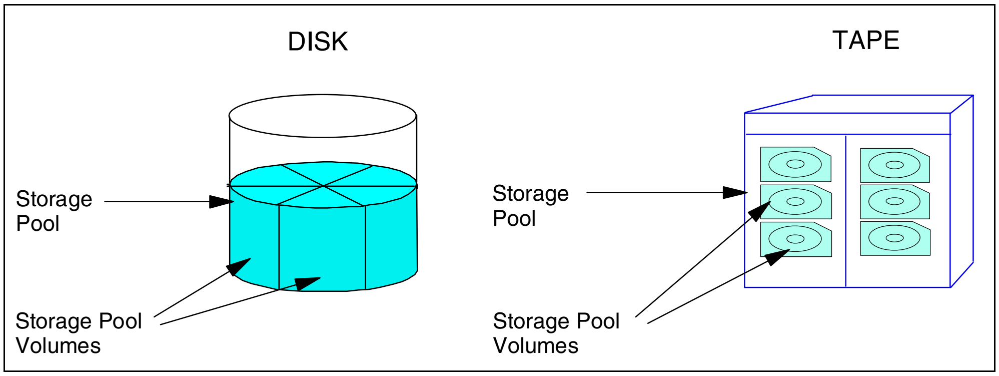


Figure 58. How volumes and storage pools are arranged for disk and tape

You can add or remove volumes without interrupting server operations. In this way you can increase or decrease the size of a storage pool dynamically without any interruption to the Tivoli Storage Manager service.

The two main categories of devices supported for storage pools are random access and sequential devices.

1. Random access devices refer to magnetic disks.
2. Sequential devices usually refer to tape devices and/or optical devices that require manual configuration. It is also possible to configure disk in a way that it will be accessed sequentially (a FILE device class).

Storage pools in Tivoli Storage Manager are defined on a wide range of different devices that may be attached locally to the server or accessed via a Storage Area Network (SAN).

For a complete up-to-date list of different devices that can be attached to the server platform that you are going to use, the best information source is the Tivoli Storage Manager Web site. This can be found at the URL:

```
http://www.tivoli.com/support/storage_mgr/adsercli.htm#servreq
```

Find your Tivoli Storage Manager server operating system type and click on the link to get to the most current device support list. Current SAN supported devices and configuration can be viewed at

```
http://www.tivoli.com/support/storage_mgr/san/overview.html
```

Tivoli Storage Manager has two types of storage pools:

- Primary storage pools
- Copy storage pools

5.1.1 Primary storage pools

When a client node backs up, archives, or migrates data, the data is stored in a primary storage pool.

When a user tries to restore, retrieve or export file data, the requested file is obtained from a primary storage pool if possible. Primary storage pool volumes are always located onsite.

A primary storage pool can use random access storage (DISK device class) or sequential access storage (for example, tape, optical or FILE device classes).

5.1.2 Copy storage pools

A copy storage pool provides an additional level of protection for client data. It is created by the administrator backing up a primary storage pool. The copy storage pool contains all current versions of all files, active and inactive, exactly as they appear in the primary storage pool.

A copy storage pool provides recovery from partial and complete failures in a primary storage pool. An example of a partial failure is a single tape in a primary storage pool being lost or defective. When a client attempts to restore a file which was on this volume, the server will automatically switch to the right copy storage volume if it is available to seamlessly restore the client’s data. If a complete primary storage pool is destroyed, for example in a major disaster, the copy storage pool is used to recreate the primary storage pool.

A copy storage pool can use sequential access storage (for example, tape optical or FILE device classes), or copy storage pools can also be created remotely on another Tivoli Storage Manager server, therefore providing electronic vaulting. See Chapter 12, “Virtual Volumes” on page 145 for more information.

Copy storage pool volumes can be moved offsite and still will be tracked by Tivoli Storage Manager. This is done by using the access mode of offsite to ensure that Tivoli Storage Manager does not request a volume mount. Moving copy storage volumes offsite provides a means of recovering from a disaster at your primary site. Tivoli Disaster Recovery Manager (see Chapter 9, “Tivoli Disaster Recovery Manager” on page 117 for more information) helps automate offsite media tracking and storage pool recovery.

Copy pools are not part of the storage migration hierarchy. Files are not migrated to or from copy storage pools. The only way to store files in copy storage pools is by using the `BACKUP STGPOOL` command.

5.2 Storage pool hierarchy

Tivoli Storage Manager allows you to configure storage pools to provide the best combination of performance throughput and data permanence. In most cases, keeping client data on tape or optical media is a requirement. However, taking the backups direct to tape may not give the best performance, especially where there are many clients to backup concurrently, and many small files are being backed up. Therefore, Tivoli Storage Manager provides a storage pool hierarchy, whereby a client initially backs up to a storage pool usually on disk. When this storage pool fills up, Tivoli Storage Manager will automatically move or 'migrate' files to the next storage pool in the hierarchy (usually on tape or optical) while the client continues its backup operation. This migration process is controlled by high and low thresholds set on the storage pool.

The management class (see Chapter 3, “Data storage policy” on page 63) determines where the client data enters the storage hierarchy. Figure 59 shows a configuration where five storage pools are defined. The default management class sends backups first to storage pool A which are then migrated by the server to pool B and finally to pool C. The other management class uses storage pool D for space managed files, whereas backups are sent directly to Tape_Pool. This would be appropriate for large client files (for example, an application database files) which can exploit the streaming performance capacity of the tape device.

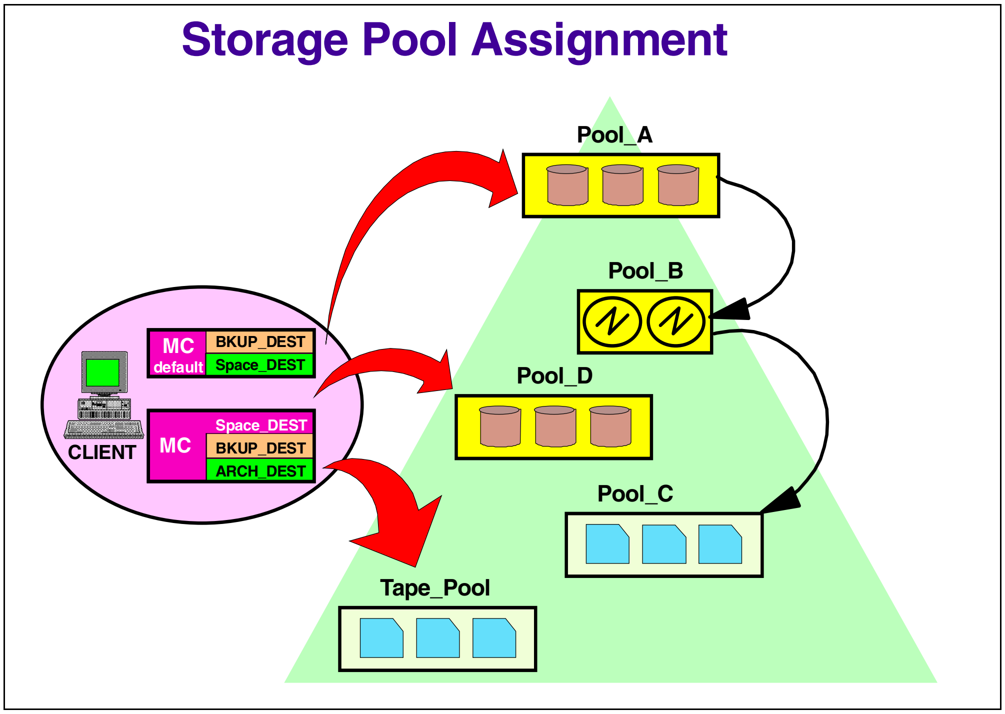


Figure 59. Possible hierarchical arrangement of different storage devices

5.3 Movement of data between storage pools

There are two controls available to help you automatically control the space in the storage pools:

- Migration
- Maxsize

5.3.1 Migration

Migration helps you control the amount of free space within a storage pool. You can define a high and a low migration threshold for each storage pool in the storage hierarchy. These thresholds tell Tivoli Storage Manager when to move data from one storage pool to another as illustrated in Figure 60.

When the amount of data in a storage pool reaches the high threshold, Tivoli Storage Manager moves client data to the next storage pool, until the storage pool reaches the low threshold. Tivoli Storage Manager's objective is to clear as much space as quickly as possible. To do this, Tivoli Storage Manager picks the client node whose collection of data occupies the most space in that storage pool and migrates the largest filespace of that client data to the next storage pool. Tivoli Storage Manager then picks the next client node that has the most data and migrates it. The migration continues until it reaches or surpasses the low threshold — and is always done at a filespace level of granularity. This operation not only clears space quickly but also keeps a client's data close together in the hierarchy. This mechanism in turn reduces the number of media mounts required during client data restoration.

For random access storage pools only, you can specify the number of processes that are used for migrating files from this storage pool. This parameter is optional and the default value is 1. You can use a migration process value equal to or less than the number of drives in the next storage pool you are migrating to; these processes are performed in parallel to improve data transfer rates. Be aware that if you have two tape drives in the storage pool you are migrating to, and if you have set this parameter to 2, this will interfere with any other requests for this tape device until the migration process is completed. If your disk pool is sufficiently sized and you have disk caching on (see 5.5.2, "Disk caching" on page 91), or a migration delay value set, then this may minimize the potential for such problems.

You can specify the number of days to delay migration for files in a storage pool: this ensures files stay in a pool for a minimum number of days. Setting this parameter may imply a larger space required for your disk storage pool, to be able to keep the files there. The benefit is that required files can be restored faster if they have not yet been migrated from the disk storage pool

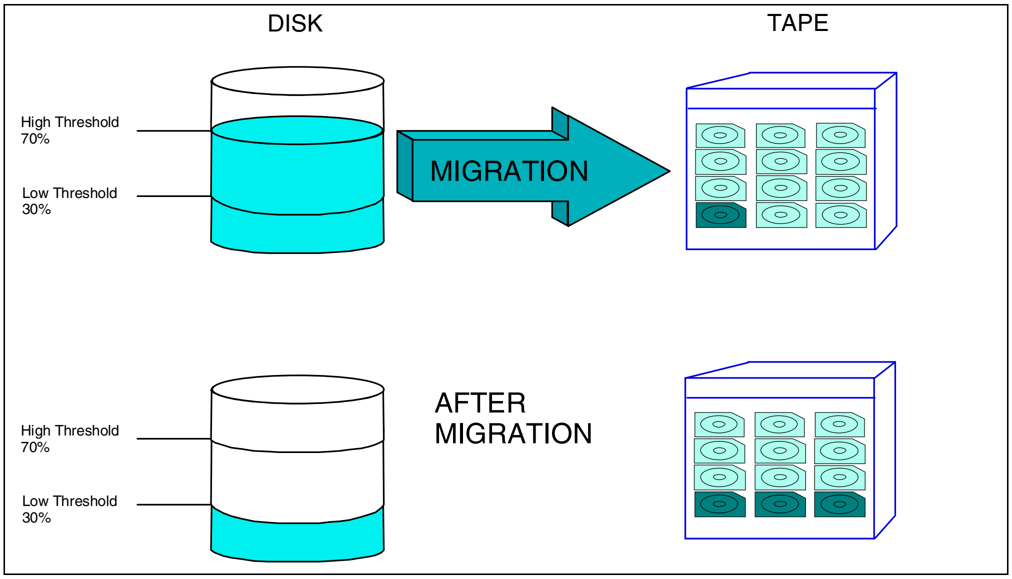


Figure 60. Migration between storage pools

5.3.2 Maxsize

If you assign a MAXSIZE value to a storage pool, files can be placed in that storage pool provided they are equal or less than the assigned MAXSIZE. If compression is used, the uncompressed size of the file is used in the comparison. If a file is too large for the first storage pool it will go straight to the next storage pool defined. This is another way to improve performance by not wasting time writing large files to the first pool, only to trigger an immediate migration when the storage pool is filled.

This parameter is not required. If you do not assign a MAXSIZE, and a file entering the storage pool causes it to exceed its high threshold, migration will automatically occur to move data to the next storage pool.

5.4 Reclamation — use less tapes

Reclamation is used to free complete tape (or optical) volumes in sequential storage pools. Because Tivoli Storage Manager keeps a defined number of versions of files as it does incremental backups, the oldest copy of a file, beyond the defined number of versions to keep, gets marked for expiry.

This file will then be deleted when the next expiration occurs. It is common that a tape volume will have files that will expire on different dates. Therefore as these files reach their expiry date and the expiration process occurs, ’virtual’ empty spaces will appear on the tape volume. Fragmentation may also appear on the volume through file deletion on the client causing the removal of many file versions because of management class definitions, or by complete filespace deletion. Fragmentation is undesirable on tapes or optical disks because it makes restores slower because of the need to skip over the empty spaces, and it also increases the total number of volumes required for data storage. It is not possible to go back and rewrite new data in these empty spaces, since these are sequential media and can only be written from the beginning to the end.

The amount of space that can be reclaimed on a volume increases as files on the volume are deleted. When the percentage of space that can be reclaimed on a volume rises above the value for the reclaim threshold set for the storage pool, that volume is reclaimed. Active files on the volume are rewritten to other volumes in the storage pool, making the original volume empty. Typically it will return to a scratch status, meaning it is then available to be reused for any Tivoli Storage Manager function requiring it. See Figure 61, which shows reclamation on two fragmented volumes to produce a 95 percent full volume and a free volume.

It is recommended that you specify a value of 50 percent or greater for this parameter so that files stored on two volumes can be combined onto a single output volume.

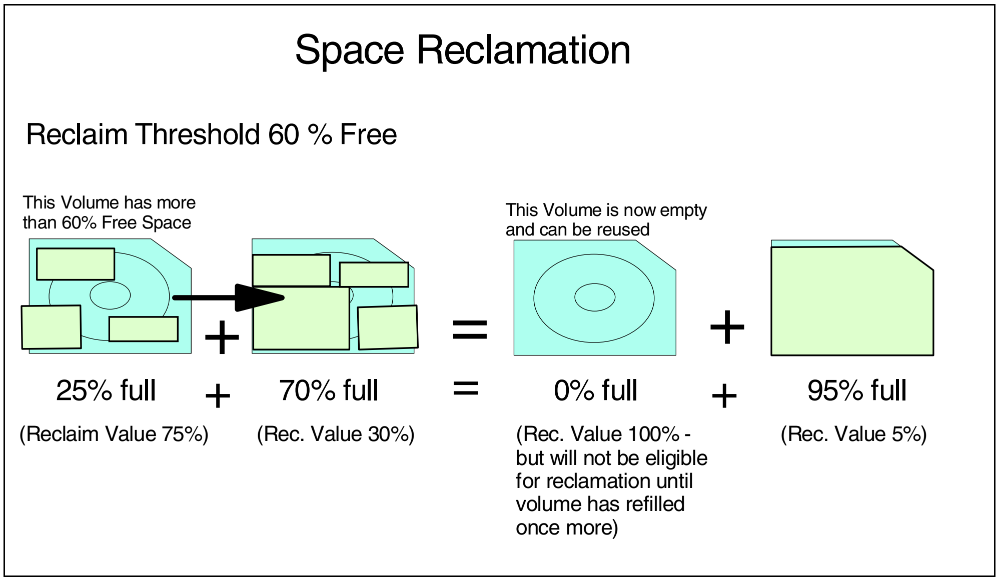

Space Reclamation
Reclaim Threshold 60 % Free
This Volume has more than 60% Free Space
This Volume is now empty and can be reused
+
=
+
25% full
+
70% full
=
0% full
+
95% full
(Reclaim Value 75%)
(Rec. Value 30%)
(Rec. Value 100% - but will not be eligible for reclamation until volume has refilled once more)
(Rec. Value 5%)

Figure 61. Reclamation on two fragmented volumes

For volumes that are not marked as offsite volumes, reclamation can occur only after the volume has been filled, and then begins to empty because of file deletion. The reclamation process is very device intensive, as it normally requires at least two available drives in the library. This might unnecessarily delay other critical Tivoli Storage Manager operations (for example, a client restore) if all drives were engaged in reclamation. So, we usually recommend to disable reclamation by setting the storage pool reclamation threshold to 100. At a suitable time, say on Saturday afternoons, reclamation can be enabled by running an administrative schedule to lower this threshold to a suitable value, which will automatically trigger reclamation to start. The threshold can then be reset back to 100 for normal operations during the week.

For offsite volumes, reclamation can occur regardless of whether the volume has ever been filled. An offsite volume is eligible for reclamation when the percentage of unused space on the volume is greater than the reclaim parameter value. The unused space includes both space that has never been used on the volume and space that has become empty because of file deletion. To avoid excessive reclamation on copy storage pools, it is also recommended to disable reclamation except for certain defined periods as described in the previous paragraph.

Another reason to control reclamation in offsite volumes is explored in 5.4.2, “Reclamation of offsite volumes” on page 88.

5.4.1 Single drive reclamation

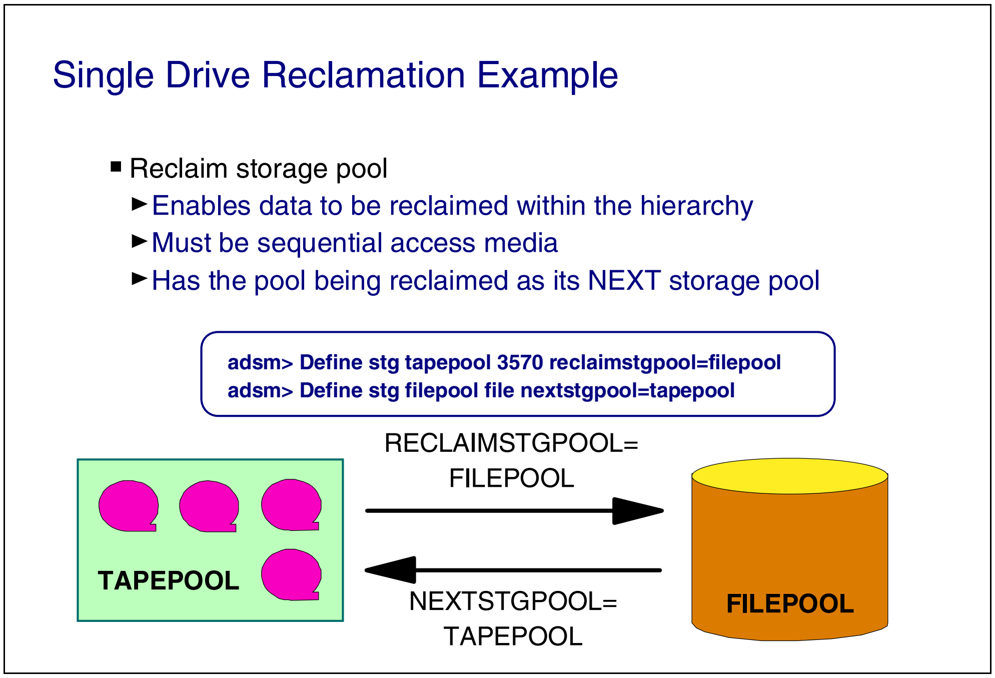
Reclamation requires two or more drives to work most efficiently. Nevertheless, reclamation can be performed by a single drive in Tivoli Storage Manager by specifying the RECLAIMSTGPOOL parameter. This parameter allows another storage pool to be used as the holding area for the sequential volume being consolidated.

The storage pool specified as the reclaim storage pool must be a primary sequential storage pool on the system. Typically, it is a new primary storage pool created for this purpose. Since it must be a sequential disk, in order to use disk as a staging area, a devicetype of FILE is required.

When the amount of reclaimable space on a media volume exceeds the reclamation threshold, Tivoli Storage Manager automatically begins the reclamation process. The volume to be reclaimed is mounted in the drive, and the active data is moved to the reclaim storage pool. If the reclaim storage pool is filled, the volume being reclaimed is dismounted, and a new volume in the same tape pool is mounted. The reclaimed data in the reclaim pool is then migrated to that tape volume. Once this process is complete, it repeats until all valid data has been reclaimed from the source volume being reclaimed.

If the reclaim storage pool is not filled before the source volume is emptied, another source volume is mounted, and reclamation continues. This process continues until either the reclaim storage pool is filled or all expired data within the storage pool has been removed.

When defining the reclaim storage pool you must also define the NEXTSTGPOOL parameter pointing back to the pool being reclaimed. Thus the reclaimed data can be migrated back to the original storage pool. See Figure 62.



Single Drive Reclamation Example

- Reclaim storage pool
  - Enables data to be reclaimed within the hierarchy
  - Must be sequential access media
  - Has the pool being reclaimed as its NEXT storage pool

```
adsm> Define stg tapepool 3570 reclaimstgpool=filepool
adsm> Define stg filepool file nextstgpool=tapepool
```

Figure 62. Single drive reclamation example

In the example shown in Figure 62, TAPEPOOL is a storage pool defined within a single drive library. TAPEPOOL is define with FILEPOOL as its RECLAIMSTGPOOL, and the FILEPOOL is defined with the TAPEPOOL as its NEXTSTGPOOL storage pool. We should note that single drive reclamation is itself a time-consuming process, so to avoid having to do this we would always strongly recommend Tivoli Storage Manager configurations with two or more drives in each library.

5.4.2 Reclamation of offsite volumes

Care must be taken in reclaiming offsite copy storage volumes.

Tivoli Storage Manager cannot physically move the data from one of these volumes to another, because they are in a vault, not available in the library. Tivoli Storage Manager manages reclamation for an offsite copy pool by obtaining the active files from a primary storage pool or from an onsite volume of a copy pool. These files are then written to a new volume in the copy pool, and the database is updated. A message is then issued that an offsite volume was reclaimed. The new volume will get moved to the offsite location, and the offsite volumes, whose active data it has combined, will be moved back to the scratch pool onsite, when the reuse delay parameter has been satisfied. See Figure 63.

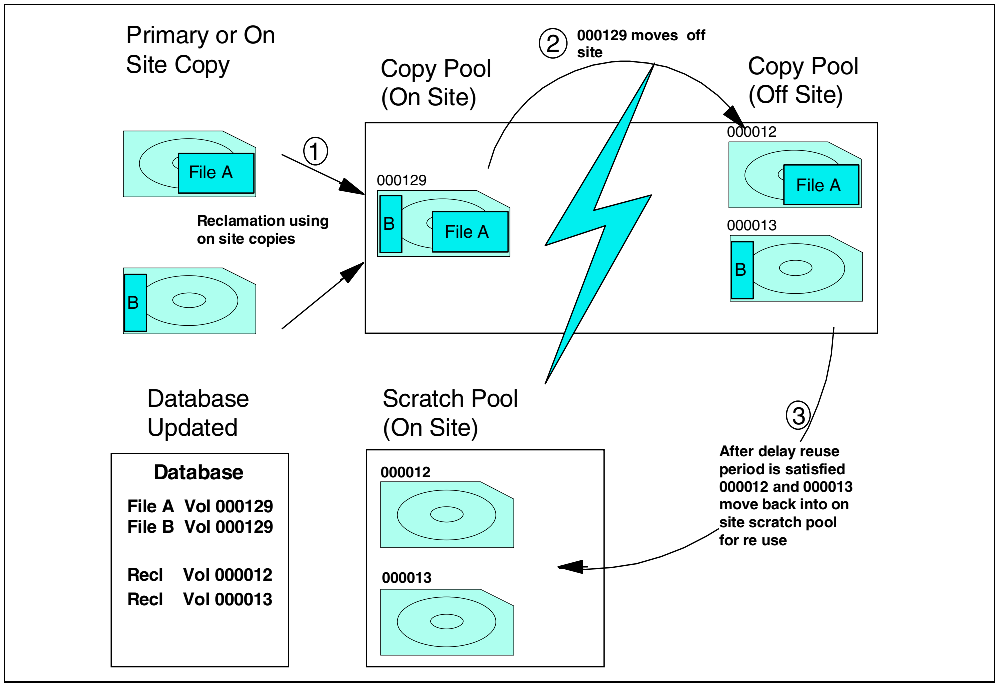


Figure 63. Reclamation of offsite volumes

However, if a disaster occurs, we could have a problem. If the reclaimed volumes have already been returned to the scratch pool and the new volume has not yet been taken offsite, we have lost any offsite backup. This is the reason for the delay reuse period for tapes returning to the scratch pool. Also, be aware that to use the offsite tape volumes 000012 and 000013 in the event of a disaster, you need the database backup that pointed to the files on these volumes.

This is achievable by controlling when reclamation will occur by scheduling it properly. Basically, you must ensure that you have performed a backup on all the onsite storage pools to their offsite copy pools, then perform a database backup so the offsite database copy points to all the files in their new locations and then you can perform reclamation. The reuse delay period defined on the storage pools must be long enough to ensure the reclaimed volumes do not return to the scratch pool before another database backup is taken.

5.5 Reduce restore times

Storage pool configuration can reduce restore times by two techniques:

- Collocation by minimizing the number of tape volumes used to store a client's data.
- Disk caching by restoring data from disk even if it has already been migrated.

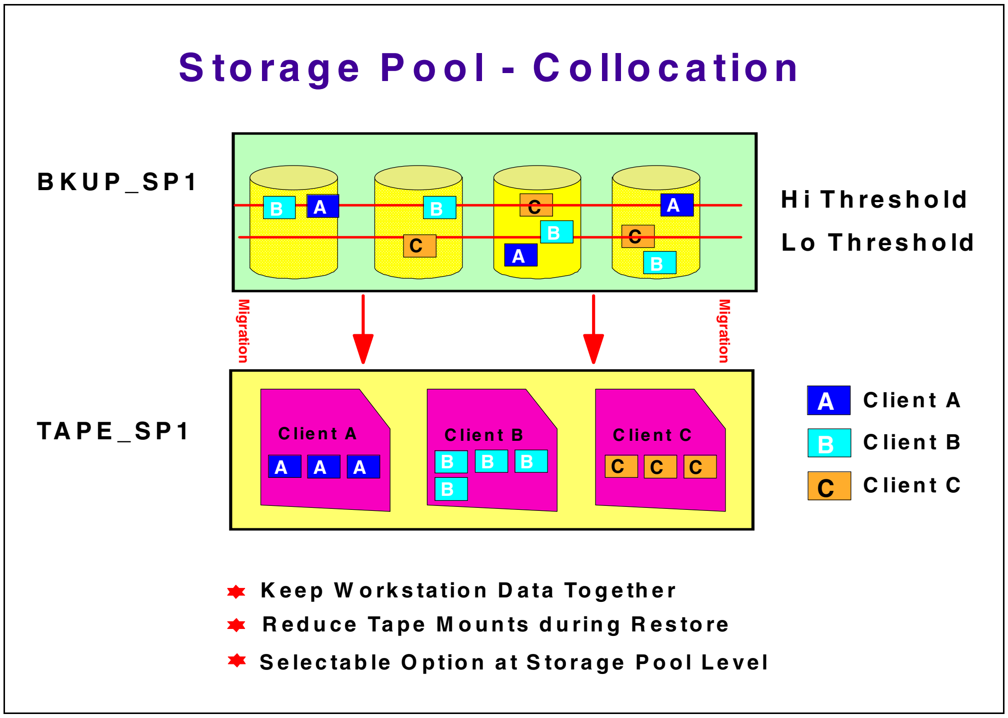
5.5.1 Collocation

Collocation is a process in which the server attempts to keep all files belonging to a client node on a minimal number of sequential access storage volumes.

To collocate data when files from different client nodes are mixed in the same storage pool, set collocation to yes when you define or update a sequential storage pool. By using collocation, you reduce the number of volume mount operations required when users restore or retrieve many files from the storage pool. Collocation thus improves access time for these operations. See Figure 64.

You can choose to have the Tivoli Storage Manager server collocate at the client or filespace level. To have collocation set at the filespace level, set collocate to filespace. If this is defined, the server attempts to put data for one node and filespace on one volume. If a node has multiple filespaces, the server attempts to place data for each filespace on different sequential volumes in the storage pool.

When collocation is disabled, the server attempts to use all available space on each volume before selecting a new volume. While this process provides better utilization of individual volumes, user files can become scattered across many volumes, so to restore a client completely may require many volume mounts.



Storage Pool - Collocation

- Keep Workstation Data Together
- Reduce Tape Mounts during Restore
- Selectable Option at Storage Pool Level

Figure 64. Storage pool collocation on a client level

Collocation will also mean that Tivoli Storage Manager will attempt to use at least one tape volume for each client. If you have a limited time for backup, more media mounts will be required with collocation, which may not be desirable. If you first back up to a disk pool and then migrate to a sequential pool, collocation works well, because of the way migration works on a client-by-client level. Collocation will also require more volumes, as well as capacity to store these additional volumes in an automated library.

However, restore times are usually critical, and collocation can provide a significant time saving. Another benefit of collocation would be to avoid the situation where two clients attempt to recover a file that resides on the same tape volume.

There are special considerations when using collocation on copy storage pools. Collocation typically results in a partially filled sequential volume for each client.

This may be acceptable for primary storage pools because these partially filled volumes remain available and can be filled during the next migration process. However, for copy storage pools this may be unacceptable because the storage pool backups are usually made to be taken offsite immediately. If you use collocation for copy storage pools, you will have to decide whether the overhead of taking more partially filled volumes offsite and increasing the reclamation activity is worth the benefit of recovering your most important clients quickly in the event of a disaster.

5.5.2 Disk caching

Disk caching is used for disk storage pools only. When cache is enabled, the migration process leaves behind duplicate copies of the files on disk after the server migrates these files to subordinate storage pools in the storage pool hierarchy. The copies remain in the disk storage pool, but in a cached state, so that subsequent retrieval requests can be satisfied quickly. However, if space is needed to store new data in the disk storage pool, the space occupied by cached files can be immediately reused for the new data. When space is needed, the server reclaims space by writing over the cached files. Files that have the oldest retrieval date and occupy the largest amount of disk space are overwritten first.

When caching is used, the occupancy value that the server compares against the migration thresholds does not include space occupied by cached copies of files. The space utilization of the pool, however, does include the space used by any cached files in the storage pool. If you update a storage pool from cache=yes to cache=no, the cached files will not disappear immediately. The occupancy value %util remains the same. The cache space will be reclaimed over time as the server needs the space, and no additional cached files will be created.

Do not use cache if you have limited database space. When you use cache, more database space is needed because the server has to keep track of both the cached copy of the file and the new copy in the subordinate storage pool.

5.6 Movement of data between Tivoli Storage Manager servers

Tivoli Storage Manager allows you to move its data from one server to another. This function is called export/import, and allows you to transfer clients between servers (for load balancing), or to move from one server operating system platform to another.

5.6.1 Export/Import

You can export or import administrator, node, server or policy information. You can export the complete server or just parts of it, for example a few clients and their stored data. See Figure 65.

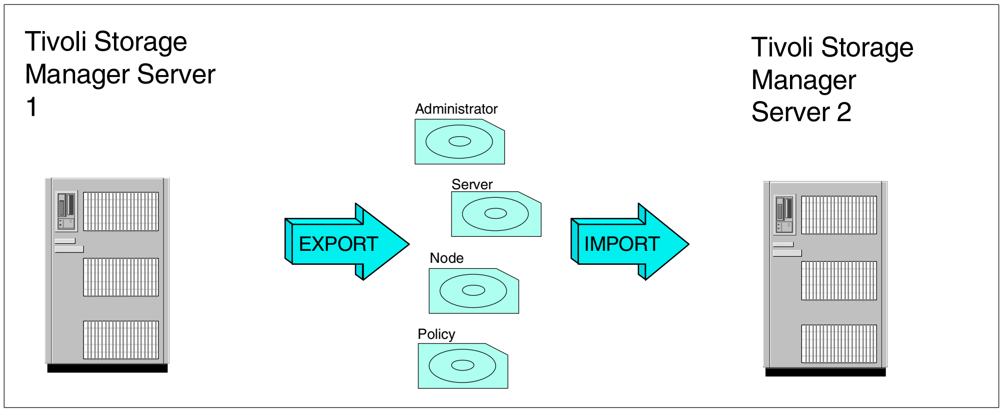

Tivoli Storage Manager Server 1
Administrator
Server
EXPORT
Node
Policy
IMPORT
Tivoli Storage Manager Server 2

Figure 65. Export data from server 1, use volumes created, import data to server 2

The export commands create an operating system-independent, self-describing copy of specified server information. The original database is not required to recover data from this volume, and Tivoli Storage Manager does not keep track of file expiration, so the information contained can be recovered onto any server at any time. This is no substitute for disaster recovery, as the number of tapes used would spiral without check. It is also a relatively time consuming process, so its designed primarily for one-off movement of data. The output media can be a tape or optical disk, a sequential volume flat file (devtype=FILE) or a virtual volume (see Chapter 12, “Virtual Volumes” on page 145). You can export configuration information as well as the backed up and archived files stored in the storage pools.

You can import to the same server platform or a different one. The server to which you are importing must support the same tape format as the server from which you exported, if you are exporting to tape media.

5.6.1.1 Export/Import Admin

These commands move administrator information such as name, password and privilege classes.

5.6.1.2 Export/Import Node

These commands move client node definitions. Each client node definition includes the user ID, password, name of the policy domain to which the client is assigned, file compression status, and backup/archive delete authority.

5.6.1.3 Export/Import Policy

These commands move policy information from one or more policy domains. They include data such as policy domain and set definitions, management class definitions, backup, copy group and archive group definitions, schedule definitions for each policy domain and client node associations.

5.6.1.4 Export/Import Server

These commands move all or part of the server control information and client file data (if specified). This includes: administrator definitions, client node definitions, policy definitions and schedule definitions defined for each policy domain. They can optionally include: file space definitions; access authorization information; and backed up, archived, and space-managed files.

5.7 Disk storage protection

Disk technology continues to improve, in terms of capacity, speed, reliability, sharability and availability. Many businesses use enterprise storage servers, which can be attached directly, via network or using a SAN. Tivoli Storage Manager exploits the services provided by this technology, to provide greater protection for itself and its client data. Disks are used within Tivoli Storage Manager for its database, recovery log and random access storage pools.

5.7.1 RAID

To protect data on a disk-type storage device, we recommend that at least you use some form of RAID, which can be implemented at either a hardware or software level.

Protecting data stored on a disk is a subject in itself, and below we just touch on some of the possibilities you may want to look into further. The best solution for you will depend on many factors:

- Server hardware and hardware RAID solutions available
- Operating system type and software RAID solutions available
- Price, performance, and redundancy level of the solution

A hardware RAID solution can be beneficial in order to reduce the performance penalty on the server, when implementing RAID. It can also provide disks and controllers that can be exchanged on the fly, should they fail (also known as hotswapping or hotplugging). Dual pathing, power redundancy, and hot spare features are other possibilities. You may also consider implementing managed server storage shared among a number of servers with dynamic allocation of shared storage.

A software RAID solution can provide levels of redundancy with a much lower initial outlay if implemented carefully, using a number of physical disks.

There are also different ways of transmitting data to the physical disks, such as SCSI, SSA, and fiber, which all have advantages and disadvantages.

Whichever path you choose, we recommend that you implement one of the three following RAID levels:

- RAID 1 Mirror
- RAID 1 + 0 Mirror and Stripe
- RAID 5 Distributed Parity

5.7.2 RAID 1

RAID 1, or Mirror, uses two physical disks and writes the same data to each of them. Should one fail, the other is an exact copy and the data is not lost—see Figure 66.

This will mean that usable space will be half of your total disk space. If you have 2 x 4 GB drives mirrored, then your logical drive will only appear as 4 GB. There is also a performance penalty, as every write to disk will now incur two writes. However, there can be an improvement on reads from disk, as the data requested can be accepted from whichever drive can satisfy the request first.

Should a drive fail, there is almost no performance penalty, except while the rebuild is being done. This might be automatic (if hot spares are available) or require the installation of a new disk and a re-synchronization of the mirrors.

Note that Tivoli Storage Manager provides its own software mirroring function specifically for database and recovery log volumes. Each volume can be mirrored up to 3-way. It is recommended to use this in-built mirroring function rather than hardware or operating system software mirroring where greater availability for the database and recovery log is required.

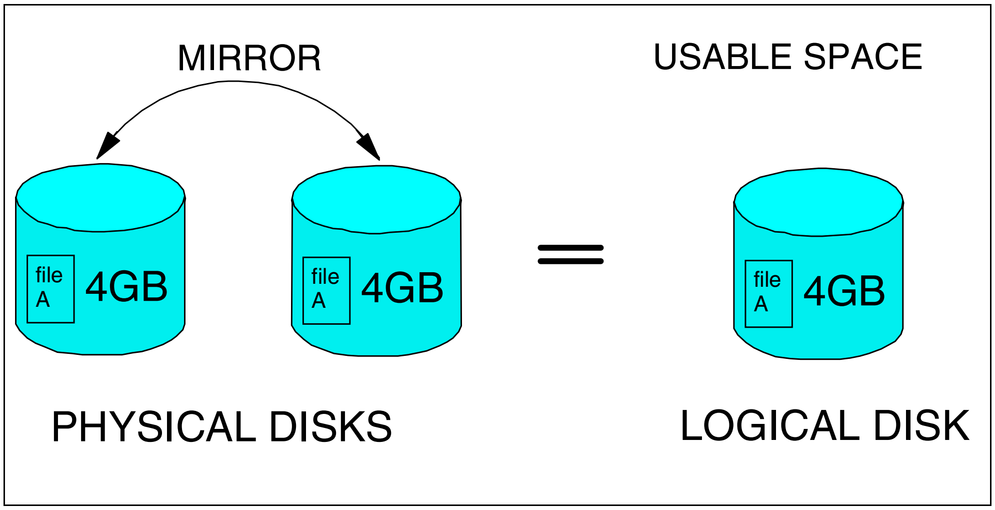


Figure 66. RAID 1 (mirroring) user usable space = 1/2 total disk space

5.7.3 RAID 1 + 0

RAID 1 + 0, or Mirror and Stripe, takes mirroring a stage further by striping the data across a number of mirrored drives. See Figure 67. RAID 1 + 0 requires an even number of physical disks, A minimum of four, with the maximum usually dependent on hardware. As all disks are mirrored initially, then should any disk fail in a pair, the data is safe. RAID 1 + 0 can withstand one disk from every pair failing. But if both disks in any pair should fail, the data is lost.

These mirrored pairs of disks are written to in turn by the controller, with blocks or chunks of data at a time. For example, with six physical disks there would be three mirror sets. The controller would write a chunk of data to the first mirrored pair, then write another chunk to the second pair, then the third pair, and the fourth chunk of data would be written on the first mirrored pair again. Refer to Figure 67.

This way of writing the data across the disks is called *striping*; it has performance advantages. Physical disks have a small amount of memory called cache, where they can store data more quickly before the slower process of actually writing it onto the disk platter. Striping makes use of this characteristic; by writing data in chunks from disk to disk, it ensures that it has given the first disk time to complete its write cycle and empty its disk cache. It can then write the fourth chunk of data straight into the disk cache again without having to wait for the disk's write cycle to complete.

Your logical drive or usable space will be half of your total disk space, but performance of this solution, potentially, will be better than RAID 1 or RAID 5.

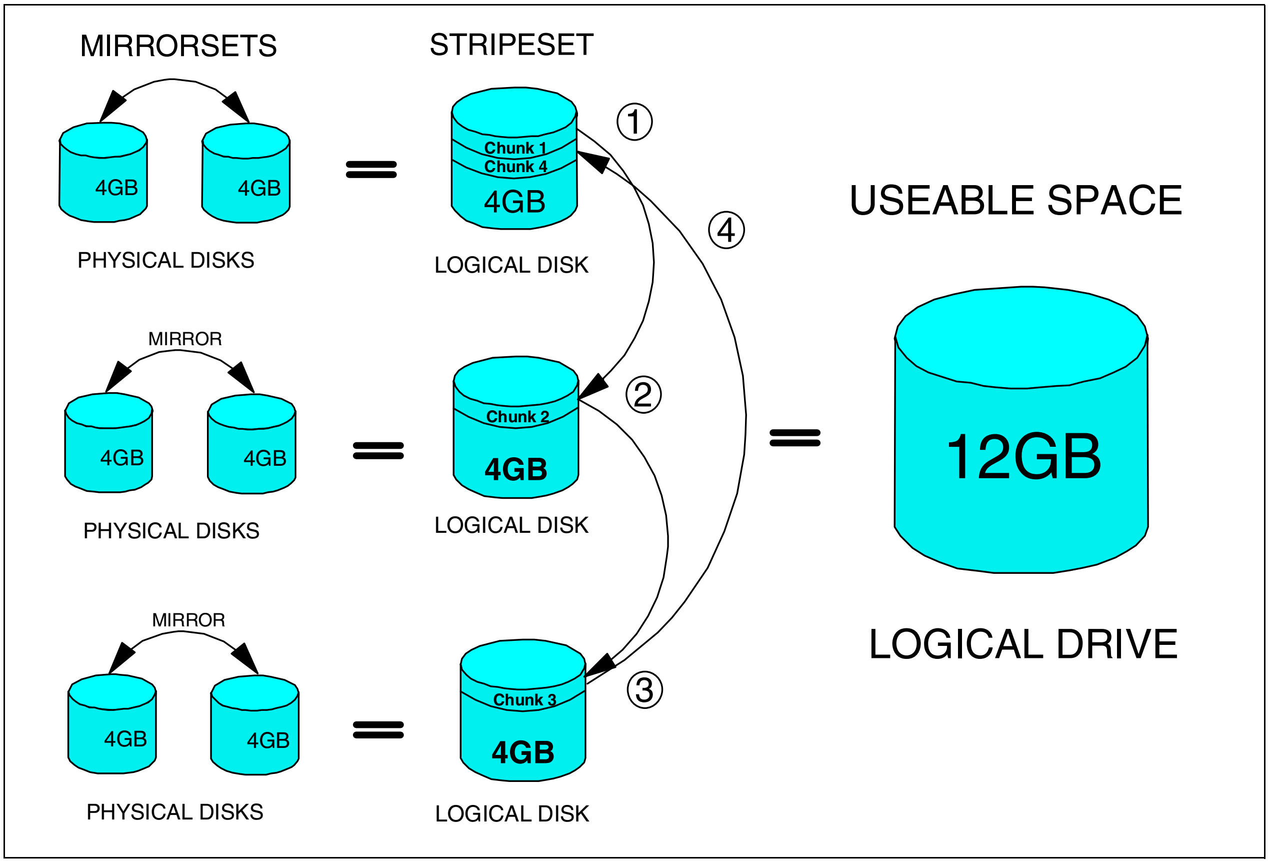


Figure 67. RAID 1 + 0, mirror and stripe

5.7.4 RAID 5

RAID 5, or distributed parity, uses three or more physical disks in a *stripeset*. See Figure 68.

It differs from the stripeset in RAID 1 + 0, because it writes a parity check on one of the disks and the data on the others.

Parity is what a RAID 5 set uses to reconstruct the data should one of the disks fail. In its simplest form it performs addition on the data of each data disk and indicates whether the result was an even or an odd number; it stores this on the parity disk. If one of the data disks fails, it can then work out what the data on the failed disk should be. See Table 5 for details.

Table 5. Reconstructing a failed data disk from the parity disk

STRIPE	DATA DISK 1	DATA DISK 2	DATA DISK 3	RESULT OF ADDITION	PARITY DISK	TO MAINTAIN PARITY	DATA 2 MUST EQUAL
1	1	0	1	2	EVEN	1+?+1=EV	0
2	1	1	0	2	EVEN	1+?+0=EV	1
3	1	1	1	3	ODD	1+?+1=OD	1
		DISK FAILS				AFTER DISK FAILS	CALC. FROM PARITY DISK

RAID 5 distributes this parity on a different disk for each stripe so that the same disk is not constantly being written to, every time parity is updated. This can improve performance over RAID levels 3 or 4.

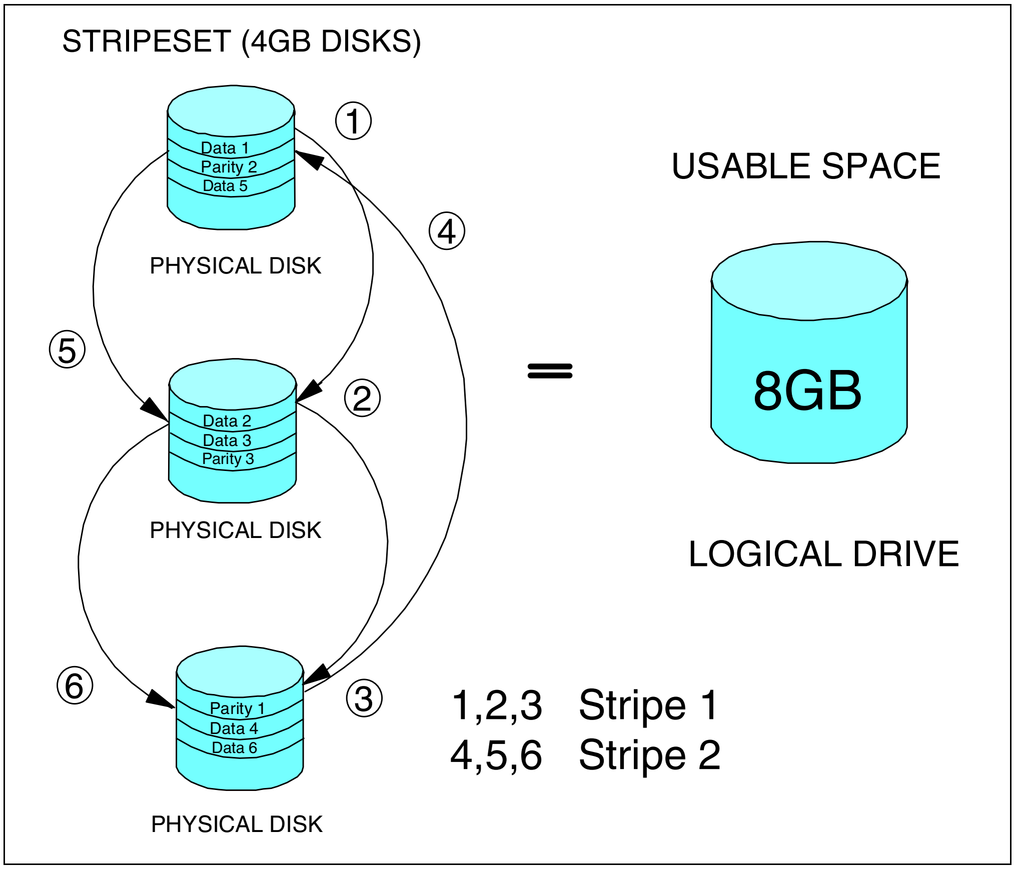


Figure 68. RAID5 stripe showing how parity is distributed

Striping the data has a benefit for performance because of cache on the physical disk, as described in RAID 1 + 0. However, performance is not quite as good as RAID 1 + 0 because of the overhead involved in calculating parity. A hardware solution can take this overhead away from the server by performing the calculations on the controller. This is especially important if a disk should fail, or if a failed disk is replaced and the RAID5 set needs to reconstruct. Most enterprise disk subsystems also have large separate write caches to improve performance.

The usable space in a RAID array can be calculated by removing the capacity of one of the physical disks from the total disk space. In Figure 68 there are 3 x 4G B disks with a total disk space of 12 GB; removing the capacity of one of them gives you a usable space of 8 GB. If there were 8 x 4 GB disks with a total disk space of 32 GB you would follow the same rule to work out your usable space. Removing the capacity of one of them (4 GB) gives you a usable space of 28 GB. This explains why RAID5 is widely used; it can be a more cost-effective solution.

Chapter 6. Tivoli Storage Manager licensing

This chapter focuses on the different features of Tivoli Storage Manager that need licensing to function. Compliance with the Tivoli Storage Manager licensing terms ensures proper system operation.

6.1 Licensed features

The base Tivoli Storage Manager server license supports an unlimited number of administrative clients, enterprise administration, one backup-archive client which can only be installed on the server itself, server to server virtual volume support for primary storage pools, and a selection of removable media devices. For current information about supported clients and devices:

- Check with your authorized reseller
- Visit the Tivoli Storage Manager page on the World Wide Web at this address: `http://www.tivoli.com/products/index/storage_mgr`

You must register additional licenses if you want to add support for any of the following optional licensed features which are not included as part of the base license agreement:

- Additional backup-archive clients. The base Tivoli Storage Manager license supports one backup-archive client
- Network Enabler to support backups of network attached clients
- Extended Device Support
- Open System environment

The following are complementary products of Tivoli Storage Manager requiring Tivoli Storage Manager as an installation pre-requisite and a separate license to use the product. The license must also be registered on the server.

- Tivoli Space Manager
- Tivoli Disaster Recovery Manager
- Tivoli Data Protection for Application Clients

The enrollment certificate files for all Tivoli Storage Manager licenses are on the Tivoli Storage Manager installation CD-ROM. You register those licenses that you are entitled to by issuing the REGISTER LICENSE command with the name of the enrollment certificate file. Once registered, the licenses are stored internally to the server.

Most licenses are platform specific, so the licenses that are to be purchased will depend on what platform you are run your Tivoli Storage Manager server. For example, if the server is on AIX then you would buy they UNIX specific licenses.

6.2 License compliance

If license terms change (for example, if a new license is defined for the server), the server conducts an audit to determine if the current server configuration conforms to the license terms.

The server also periodically audits compliance with the license terms. The results of this audit are used to check and enforce license terms. If 30 days have elapsed since the previous license audit, the administrator cannot cancel the audit.

If the server uses a licensed feature but the license is not registered, the function fails. When you issue a command associated with an unlicensed feature, Tivoli Storage Manager does not issue a warning message, and the command fails.

6.3 Tivoli Storage Manager licenses

To obtain licenses for licensed features and complimentary products, you can register the following licenses:

1client.lic

- One additional backup-archive client

5client.lic

- Five additional backup-archive clients

10client.lic

- Ten additional backup-archive clients

50client.lic

- Fifty additional backup-archive clients

  The client licenses are registered to the Tivoli Storage Manager Server platform configuration that you setup. You do not need to specify which operating system platform each client uses. The client code to be installed on the clients (nodes) is free but the number of licensed nodes is dependent on the number of licenses registered on the server. Tivoli Storage Manager has one free client license, so if you buy 50 licenses, then you have a total of 51 client licenses.

network.lic

This is required if you are using a network communication method, for example, TCP/IP to connect to the server.

A network license is not required for the following:

- Administrative client Web interface (HTTP communication method)
- Administrative client GUI or command line access
- The use of a Tivoli or an SNMP receiver
- A client on the same physical system as the Tivoli Storage Manager server using the SHARED MEMORY or NAMED PIPES communication method.

opensys.lic

- OpenSystem environment support—this license gives you the ability to back up the Unix System Services portion of OS/390 and also AFS/DFS filesystems where supported.

advdev.lic

Advanced device support:

- License for devices that are recently added to the list of Tivoli Storage Manager devices. See the Tivoli Storage Manager supported devices list at the Web site: `http://www.tivoli.com/support/storage_mgr/adsercli.htm` for the devices which need the Advance Device License.

spacemgr.lic

Tivoli Space Manager (formerly HSM client)

- This complementary product maximizes usage of existing storage resources by transparently migrating data off client hard drives based on size and age criteria to the Tivoli Storage Manager server, leaving only a stub file. When the migrated data is accessed, Tivoli Space Manager transparently migrates the data back onto the local disk from the server. Licensing this feature enables the Tivoli Storage Manager to support space managed clients. Not all Tivoli Storage Manager clients have a space management function - check on the Tivoli Storage Manager Web page for currently supported clients.

drm.lic

Tivoli Disaster Recovery Manager:

- Automatically generates a disaster recovery plan—containing the information, scripts, and procedures needed to automate restoration—that helps ensure quick recovery of your data after a disaster.
- Automatically manages and tracks the media on which your data is stored—whether onsite, in-transit, or offsite in a vault—so your data is easily located if disaster strikes.
- Includes additional virtual volume support for copy storage pools and server database backups.

oracle.lic

Tivoli Data Protection for Oracle

- Enables centralized, on-line, incremental, back-up capabilities, and automated storage management function of Oracle databases using the Oracle Enterprise Backup Utility (EBU) or Oracle8 Recovery Manager (RMAN).

informix.lic

Tivoli Data Protection for Informix

- Enables backup and restore of Informix databases and logical logs. It uses the Informix On-Bar utility to perform the operations.

sapr3.lic

Tivoli Data Protection for R/3

- Enables data backup and recovery of R/3 databases from SAP.

mssql.lic

Tivoli Data Protection for MS SQL Server

- Enables centralized complete and incremental online backup of SQL Server databases with Tivoli Storage Manager. The utilities provide full online backup and restore of all databases and transaction logs.

msexch.lic

Tivoli Data Protection for MS Exchange

- Centralized online full, copy, incremental, and differential backups of Exchange Directory and Information Stores.

lnotes.lic

Tivoli Data Protection for Lotus Notes

- Enables centralized, on-line, incremental backup of the complex logical structure of Lotus Notes databases.

domino.lic

Tivoli Data Protection of Lotus Domino

- Enables Lotus certified online backups, restore and archive of Lotus Domino R5 Server databases and transaction logs.

Chapter 7. Web administrative and client interfaces

Tivoli Storage Manager provides two Web browser interfaces into its client and administrator functions. These interfaces provide essentially the same functionality as the command-line or native graphical clients but with increased flexibility. The client web interface is for performing operations for a particular client node, such as backup and restore. The administrator web interface allows an administrator to manage and control the actual server resources.

The two interfaces are frequently confused because they have similar names, are accessed by similar means, and both use signon IDs that are defined as Tivoli Storage Manager administrators.

The web administrative interface connects to a Tivoli Storage Manager server through the Web to present an interface to a true Tivoli Storage Manager administrator session. It is used by an administrator to configure, monitor and control a Tivoli Storage Manager server just as they would use the command line interface. The browser interface has some features in common with the native Windows GUI interface, which used to be shipped with older product releases; however, this component is no longer available or supported.

The web backup-archive client connects to a Tivoli Storage Manager backup-archive client process through the Web to present an interface to remotely perform the functions of the backup-archive client. It is used by Tivoli Storage Manager administrators, help-desk personnel or end users to manually backup, restore, archive, or retrieve files on a remote client without having to physically visit the client. To enforce security, the web backup-archive client connects to the actual client node using an administrator ID with NODE authority. This eliminates the requirement for unique security definitions on every client for the Tivoli Storage Manager administrators, help desk personnel or end users. See “User management” on page 107 for more information on Tivoli Storage Manager user IDs.

7.1 Web backup-archive client

The workstation connecting to the web backup-archive client uses a Java 1.1.5-capable Web browser, such as Microsoft Internet Explorer 4.01 or Netscape 4.06. Using a Web browser means that Tivoli Storage Manager code is not required on the workstation where you are using the browser. The web backup-archive client Tivoli Storage Manager code is required on the client to which you are connecting.

This interface provides the basic backup, restore, archive and retrieve functions remotely and without authorization problems. Once connected, the interface is similar to the backup-archive client native GUI interface. It is important to note that the administrator is acting remotely only, all restores or retrieves will be stored back on the original backup client node. The web client interface does not of itself give the authority to ’steal’ files and store them on the system running the browser. This is possible only if *client owner* authority is used (see 8.1.3, “Privilege classes” on page 108 for an explanation of the different administrator authorities provided).

The following functions are not supported by the web backup-archive client: estimate, view policy information, access another user, file details, command line, preferences editor, connection information, searching and filtering.

Figure 69 shows the first screen of the Web Backup-Archive Client.

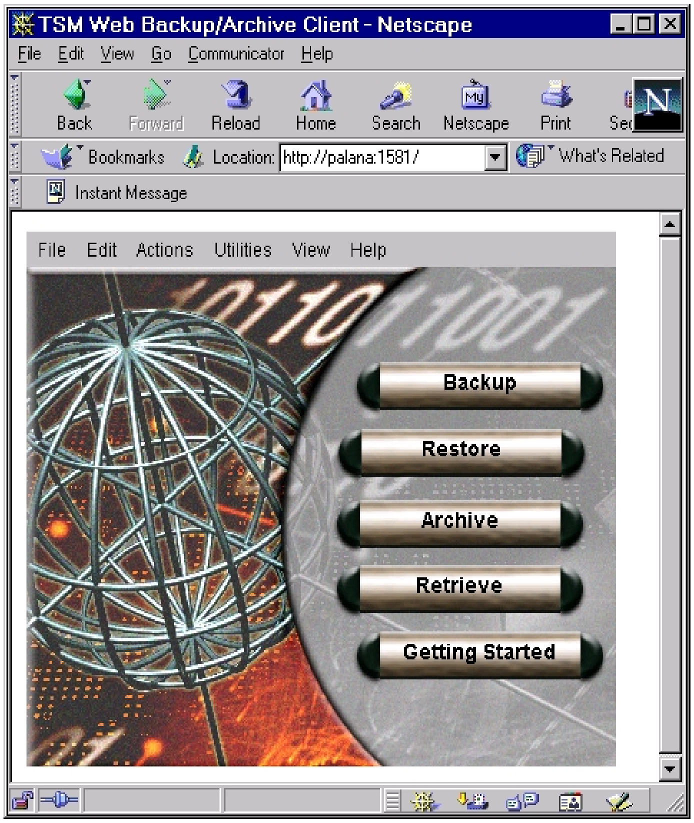

TSM Web Backup/Archive Client - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Se
Bookmarks Location: http://palana:1581/ What's Related
Instant Message
File Edit Actions Utilities View Help
Backup
Restore
Archive
Retrieve
Getting Started

Figure 69. Screen shot of the Web Backup-Archive Client

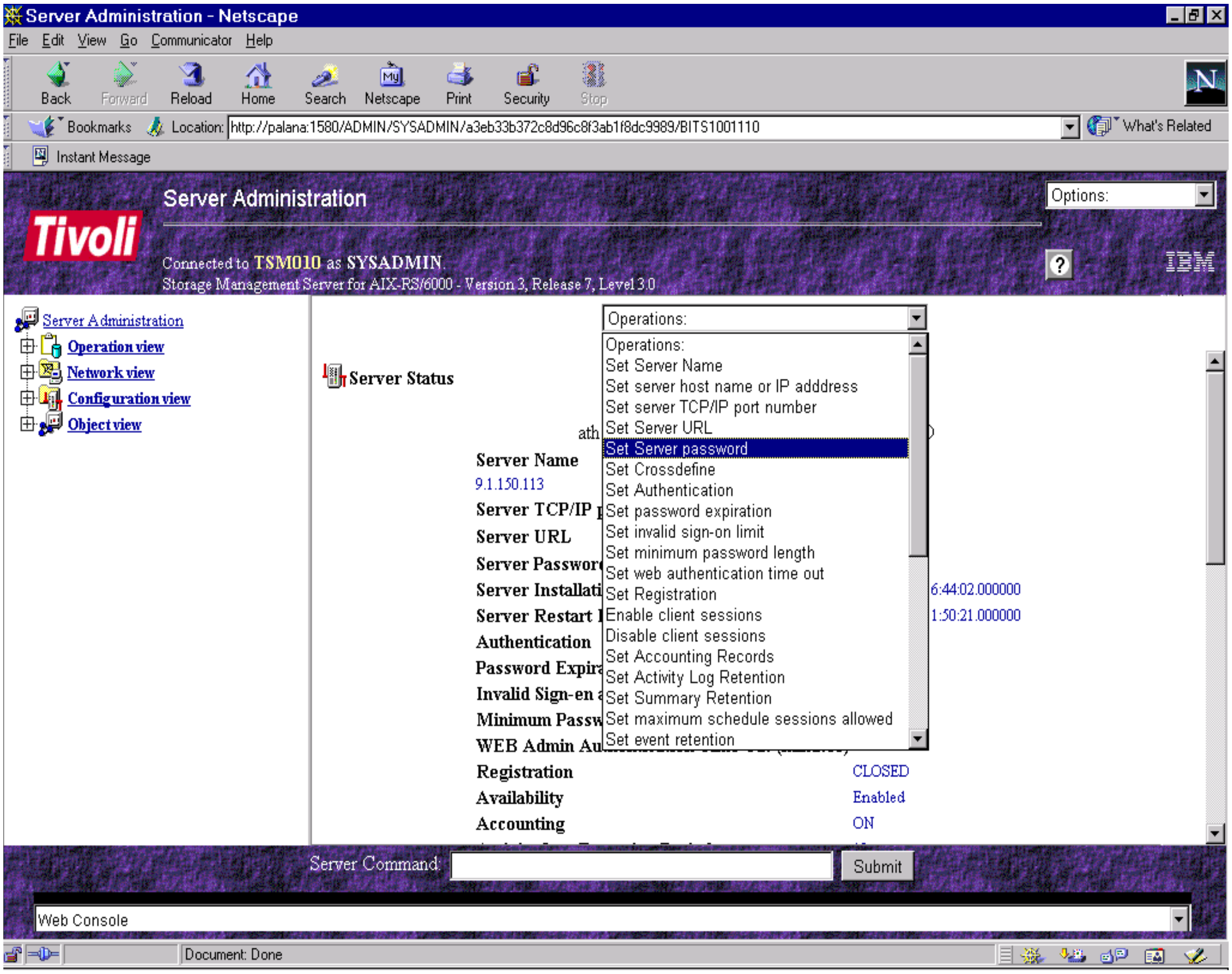
7.2 Web administrator

The web administrative interface connects to a Tivoli Storage Manager server through the Web to present an interface to a true Tivoli Storage Manager administrator session. It requires a Java 1.1.5-capable web browser such as Microsoft Internet Explorer 4.01 or Netscape 4.06. Using a Web browser means that you do not need to run any Tivoli Storage Manager code on the Web browser workstation.

As opposed to the discontinued Windows GUI interface, the web administrator now includes all the function of the command line interface. The main screens present a tree view on the left with Operation, Network, Configuration and Object views. Each clickable view opens up to present other options. Some functions can be performed in more than one area. For example, adding an administrator can be done in each of Operations and Object. In many options, a drop-down box, usually called Operations, is available to perform commands and functions relevant to this area.

The web administrative interface has a drop-down box in the upper right corner of the screen that allows you to perform three functions: show a command line, show an event line, and log off the administrator. In the command line area, you can issue any Tivoli Storage Manager command that could be issued from the command line interface. The event line displays the command issued as a result of performing a function using the web GUI. It maintains a history of the commands that have been issued in this session. Both the command line and the event line can be displayed at the same time.

Figure 70 shows the Server Status screen of the web administrative interface and an open drop-down box with the Set password expiration option selected. Also shown are the server command line and the events console window.



Connected to TSM010 as SYSADMIN.
Storage Management Server for AIX-RS/6000 - Version 3, Release 7, Level 3.0

Server Name
9.1.150.113
Server TCP/IP
Server URL
Server Passwor
Server Installati 6:44:02.000000
Server Restart I 1:50:21.000000
Authentication
Password Expir
Invalid Sign-en
Minimum Passw
WEB Admin Au
Registration CLOSED
Availability Enabled
Accounting ON

Figure 70. Server status screen of the web administrative interface

The web administrator interface is platform independent, it is location independent (you only need a Web browser), and it allows you to perform all Tivoli Storage Manager administrator functions.

Chapter 8. User management

This chapter explains the functions of Tivoli Storage Manager administrators and its client option sets.

8.1 Administrators

A Tivoli Storage Manager administrator manages Tivoli Storage Manager resources on the server such as storage pools, devices and data management policies. An administrator can also be responsible for backup and restore of client data. The number of administrators and their level of privileges will vary according to your environment. In a very small implementation, all of these functions could be performed by a single person.

8.1.1 Authority levels

Administrators with system privilege can perform any Tivoli Storage Manager function. Administrators with policy, storage, operator, analyst, or node privileges can perform subsets of Tivoli Storage Manager functions. All administrators, even those with no specific privileges, can perform Tivoli Storage Manager query functions since these need only read access.

8.1.2 Creation

There are two ways to create a Tivoli Storage Manager administrator ID: the `register node` and `register admin` commands. The `register admin` command is used to explicitly create an administrator ID with certain defined privileges. The `register node` command automatically creates an administrator ID with the same name as the node with owner access privilege to the node. This administrator ID can then be used with the client web browser interface to perform remote backup/restore operations. If an administrator with the same name as the client node already exists during registration of a new node, then this administrator ID is automatically updated to grant owner access to it.

You can optionally specify `userid=none` when registering new nodes to prevent creation of the administrative ID. Instead, the `register admin` or `grant authority` command can be used to create and grant access to specific nodes and their data. This is useful where there is a centrally located helpdesk requiring access to a number of client nodes, to assist remotely in client backup and restore functions without the need to be physically at the client location.

Administrators must be registered with a name and a password. The maximum length of an administrator name is 64 characters. The maximum length of an administrator password is 64 characters.You can optionally specify that the password be changed the first time that an administrator logs on.

After administrators are registered, they can make queries and request command-line help. To perform other Tivoli Storage Manager functions, they must be granted authority by being assigned one or more administrative privilege classes.

At installation, the server console is defined with a special user ID, which is named SERVER_CONSOLE. This name is reserved and cannot be used by another administrator. It also cannot be deleted. At installation, the

SERVER_CONSOLE user ID can be used to register other administrators and grant privileges.

8.1.3 Privilege classes

Privileges are granted to an administrator through the `grant authority` command. You need system privilege to issue this command.

Figure 71 shows the hierarchy of administrative privilege classes. Each box represents a privilege that could be given to an administrator.

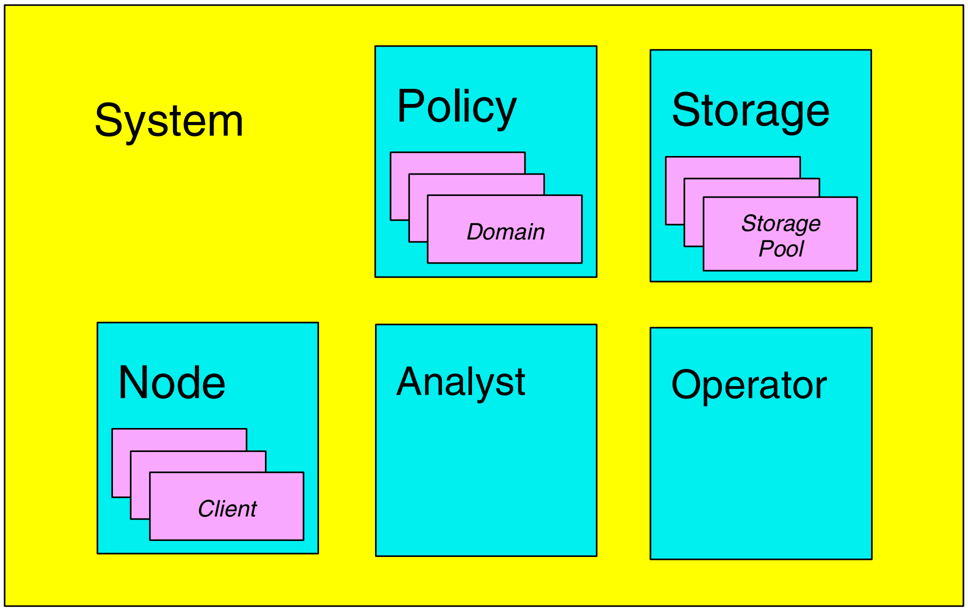


Figure 71. Administrative privileges

The *policy* privilege can be assigned for all domains (unrestricted privilege) or for one or more of the defined domains (restricted privilege). The *storage* privilege can be assigned for all storage pools (unrestricted privilege) or for one or more of the defined storage pools (restricted privilege). The *node* privilege can be assigned on an individual client basis.

8.1.3.1 System

An administrator with system privilege can perform any Tivoli Storage Manager administrative task. All other privileges are included when an administrator is assigned system privilege.

8.1.3.2 Policy

An administrator can have either unrestricted or restricted policy privilege.

An administrator with unrestricted policy privilege can manage the backup and archive policy definitions (for example, management classes, copygroups and schedules) for client nodes assigned to any policy domain. When new policy domains are defined to the server, an administrator with unrestricted policy privilege is automatically authorized to manage the new policy domains.

An administrator with restricted policy privilege can perform the same operations as an administrator with unrestricted policy privilege but only for specified policy domains.

8.1.3.3 Storage

An administrator can have either unrestricted or restricted storage privilege.

An administrator with unrestricted storage privilege has the authority to manage the Tivoli Storage Manager database, recovery log, and all storage pools. An administrator with unrestricted storage privilege cannot define or delete storage pools as this requires system privilege.

Administrators with restricted storage privilege can manage only those storage pools to which they are authorized. They cannot manage the Tivoli Storage Manager database or recovery log.

8.1.3.4 Operator

Administrators with operator privilege control the immediate operation of the Tivoli Storage Manager server and the availability of storage media.

8.1.3.5 Analyst

An administrator with analyst privilege can issue commands that reset the counters that track server statistics, but otherwise can perform only query commands.

8.1.3.6 Node

Administrators with node privilege can remotely access a Web backup-archive client and perform backup and restore actions on that client using an administrative user ID and password. The privilege can be for specific node/s or all client nodes in a domain.

Client Owner Authority

A user with client owner authority is able to access the client and its data from either a remote Web client or the native backup client so that data can be restored either to the original client or to another client.

Client Access Authority

A user with client access authority can access the client and its data from the remote Web client and can only restore the client data back to the original client.

Figure 72 demonstrates client access and owner authority.

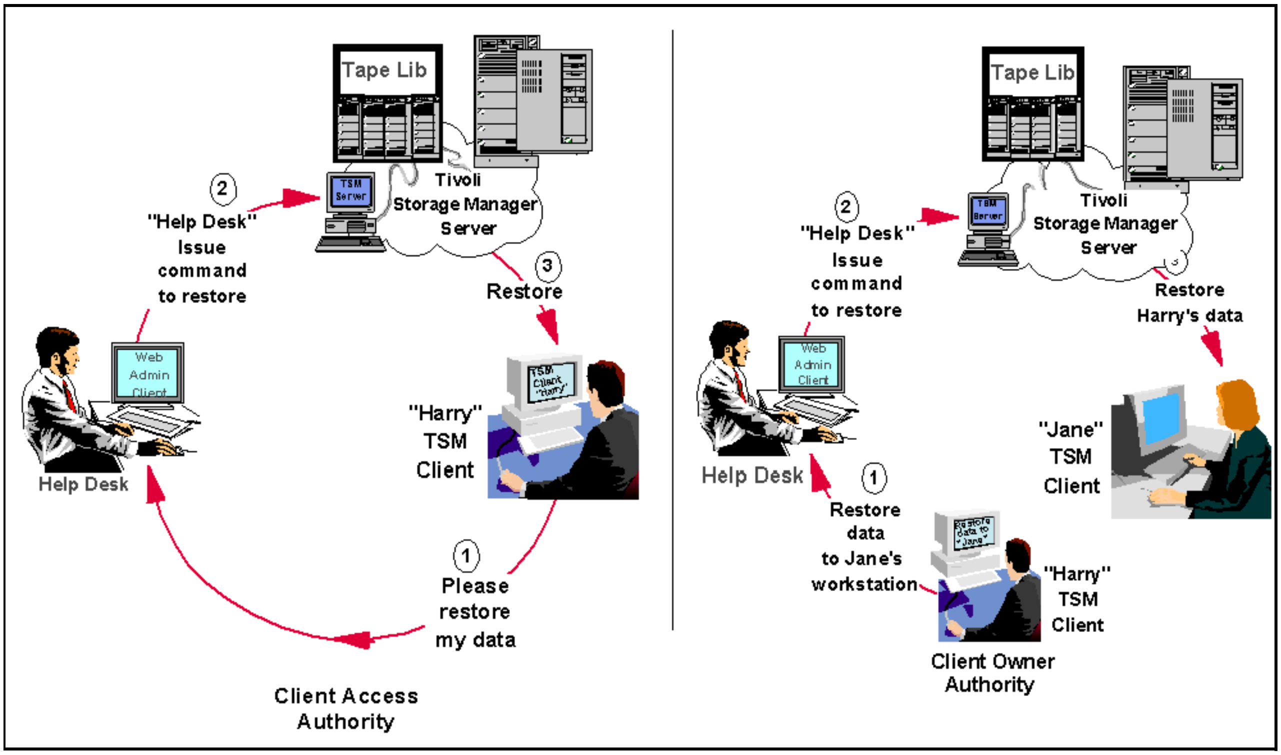


Figure 72. Client Access Authority and Client Owner Authority

8.1.4 Operations

There are a number of operations that you can perform with administrators. This section deals with those operations and some related considerations.

8.1.4.1 Renaming an administrator

An administrator is renamed through the `rename admin` command. You require system privilege to issue this command.

You can rename an administrator ID when an administrator wants to be identified by a new ID, or you want to assign an existing administrator ID to another person. You cannot rename an administrator ID to one that already exists on the system.

8.1.4.2 Changing administrative authority

You can extend, revoke or reduce another administrator's authority through the `grant authority` and `revoke authority` commands. You require system privilege to issue these commands.

Granting authority to an administrator adds to any existing privilege classes; it does not override those classes. You can reduce an administrator's authority simply by revoking one or more privilege classes and granting other classes as needed.

8.1.4.3 Removing an administrator

You can remove administrators from the server so that they no longer have access to administrator functions through the `remove admin` command. You require system privilege to issue this command.

Where there is an administrator ID with the same name as a client node, removing the client node also causes the administrator id to be removed. This is the reverse behavior as when a client node is created with default creation of administrative userid.

You cannot remove the last system administrator or the SERVER_CONSOLE administrative ID from the system.

8.1.4.4 Locking and unlocking an administrator

You can lock out administrators to temporarily prevent them from accessing Tivoli Storage Manager through the `lock admin` and `unlock admin` commands. You require system privilege to issue these commands.

You use the `lock admin` command to prevent an administrator from accessing the server. The administrator is locked out until a system administrator uses the `unlock admin` command to reestablish access for the administrator.

You cannot issue the `lock admin` command against the SERVER_CONSOLE administrative ID.

8.1.4.5 Requesting information about administrators

You can query the server to view administrator information through the `query admin` command. Any administrator can issue this command. You can request for information for one or more administrators. You can also query all administrators authorized with a specific privilege class.

8.1.5 Security

Authentication of administrators in a Tivoli Storage Manager environment is optional. The default is that authentication is required.

With password authentication set to on, all administrators must enter a password when accessing the server. With password authentication set to off, administrators can access Tivoli Storage Manager without entering a password.

Authentication is controlled by the `set authentication` command.

When security authentication is in effect, there are a number of security options that can be specified to implement a security policy. These options are related to the use of passwords. They are explained in the remainder of this section.

8.1.5.1 Maximum logon attempts

You can set the maximum number of logon attempts allowed before an administrator is locked. The server maintains a count of successive invalid password attempts. When that count reaches the maximum, that administrator is locked out until another system administrator uses the `unlock admin` command to reestablish access for that administrator. When a successful logon occurs, the count of invalid password attempts is reset to zero.

This maximum number of logon attempts option is controlled by the `set invalidpwlimit` command and it is a global setting.

8.1.5.2 Password expiry

You can set the number of days that a password is valid. When that period expires, an administrator or node will be prompted for a new password when he or she next attempts to logon.

The password expiration option is controlled by the `set passexp` command. Password expiration can be set for nodes, administrators or both.

8.1.5.3 Minimum password length

You can set the minimum length of a password. When specifying a new password or creating an administrator, you must specify a password that contains at least the number of characters specified by this option. You can also specify that there is no minimum password length.

The minimum password length option is controlled by the `set minpwlength` command and its a global setting.

8.1.5.4 Web authentication timeout

You can set the timeout interval for the Web administrative interface. After that interval has elapsed, an administrator must re-enter the administrator name and password to continue the session. The installed default is 10 minutes. You can also specify that there is no timeout value.

The Web authentication timeout option is a global setting and is controlled by the `set webauthtimeout` command.

8.1.6 Auditing

All commands issued by all administrators are logged to the server activity log. This cannot be removed or altered in any way.

The information is written to the log as a message. The message contains the name of the administrator who issued the command and the full text of the command. The command is logged regardless of whether it was valid or not. Any passwords in the command are replaced by the special string `?****?`.

8.2 Client option sets

A Tivoli Storage Manager client session has a set of options which are used during the backup, archive, restore, or retrieve processes. Options can be specified in two ways: (1) a client options file, and (2) a client options set. The first is mandatory, while the second is optional.

The client options file is a configuration file (or files, in the case of UNIX clients only) which is local to each Tivoli Storage Manager client. It contains entries of valid client options with an associated value. It also contains include-exclude file specifications. A client option set is a set of Tivoli Storage Manager client options stored in the Tivoli Storage Manager database. It is used in conjunction with a client options file. An option set can be associated with one or more clients, but a client can be associated with only one option set.

You use client option sets for ease of administration. Management of the environment is complex where the number of clients is growing and the number of options is increasing. The use of client option sets eases that administrative burden by centralizing the management of those options and clients. It is easier to update a client options set once, than to perform the same update to the local client options file on each node.

The options defined in a client option set are a subset of the available client options. Options such as communications are still stored on the client machine. When the same individual option is specified in both the local options file and the options set, the default is that the options file version is used. However, you can specify that individual options in an option set cannot be overridden in the client's local option file. Although include-exclude specifications cannot be overridden, you can specify the sequence in which the option set specifications are processed. Therefore, one set of default values can be defined for each type of client, and the client machines can still be customized, within acceptable limits.

Part 3. Advanced functions

Part 3 covers functions that may be useful, depending upon the nature of the implementation details. It builds upon the base functions covered in Part 2, “Standard base function” on page 25.

Chapter 9. Tivoli Disaster Recovery Manager

The most recent version of this material is maintained on the ITSO Internet Web pages at:

```
http://www.redbooks.ibm.com/solutions/systemsmanagement/tsm/c_drm.html
```

One of the big Information Technology challenges is to have not only complete backup of all data, but also the ability to recover it in a timely fashion. This means having controls in place to keep track of massive storage repositories in complex environments with many different machines, devices, tapes and applications. This chapter shows how Tivoli Disaster Recovery Manager helps the administrator with this formidable task.

9.1 What is disaster recovery?

Disaster recovery is the process of restoring operations of a business or organization in the event of a catastrophe. There may be many aspects related to the restoration, including facilities, equipment, personnel, supplies, customer services, and data. One of the most valuable business assets is the critical data that resides on the computer systems throughout the company. The recovery of this data needs to be a primary focus of the disaster recovery plan. Tivoli Storage Manager, along with Tivoli Disaster Recovery Manager, will assist you in the technical steps that you need to make your data available to users after a widespread failure. Some generic terms and terminologies that you may find regarding disaster recovery are:

- Business Impact Analysis (BIA): It considers business criticality for all managed systems and information flows that are vital for business survival. One of the key points here is to evaluate and map all internal departments that may have dependency to business continuity. This is an important task in a company and it is out of the scope of Tivoli Storage Manager and Tivoli Disaster Recovery Manager.
- Business Continuity Plan / Business Recovery Plan (BCP/BRP): It is the result of an analysis and the documentation on how to bring the company back to normal in an orderly way, considering its priorities and requirements based on the facilities and resources required. Some of the requirements will be translated into the Disaster Recovery Plan which may be used by Tivoli Storage Manager.
- Information Technology Recovery Plan or Disaster Recovery Plan (DRP): Defines what you have to rebuild and how to do it for the business side, such as database applications, spreadsheet data, user files and so forth. This is the component where Tivoli Storage Manager and Tivoli Disaster Recovery Manager are of principal assistance.

Figure 73 shows the relationships between these entities. As you can see, although Tivoli Disaster Recovery Manager plays an important role, there are many other issues to consider.

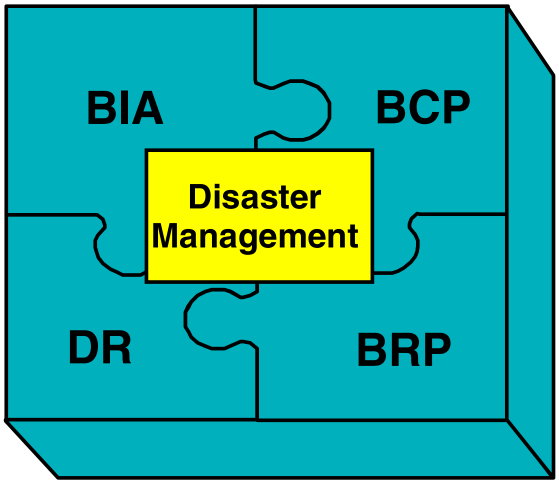


Figure 73. Disaster recovery terminology

Distributed data recovery restores data to workstations, application servers, and file servers in the event of data loss due to accidental erasure, media failures, sabotage, and natural disasters. It involves creating, managing, and recovering copies of distributed data. These copies should be taken offsite to minimize the chance that a disaster may destroy backup copies along with primary copies. Many data administrators also choose to keep backup copies onsite to expedite recovery from smaller media failures.

Disaster recovery requires creating copies of primary data at a minimum. Many businesses and backup products stop here. To achieve a complete recovery solution for distributed data, several additional features must be considered.

9.1.1 Using Tivoli Disaster Recovery Manager

Tivoli Storage Manager delivers disaster recovery as an optional product, known as Tivoli Disaster Recovery Manager (DRM), that you can license to your base product. It offers various options to configure, control and automatically generate a disaster recovery plan containing the information, scripts, and procedures needed to automate restoration and help ensure quick recovery of your data after a disaster. It also manages and tracks the media on which your data is stored, whether onsite, in-transit, or in a vault, so that your data can be easily located if disaster strikes. The scripts can help you document your basic Information Technology recovery strategy, the steps to rebuild your core systems, as well as the critical machines that you must recover.

Tivoli Disaster Recovery Manager enhances Tivoli Storage Manager tape controls and gives you extra options such as the recovery plan file, as seen on Figure 74.

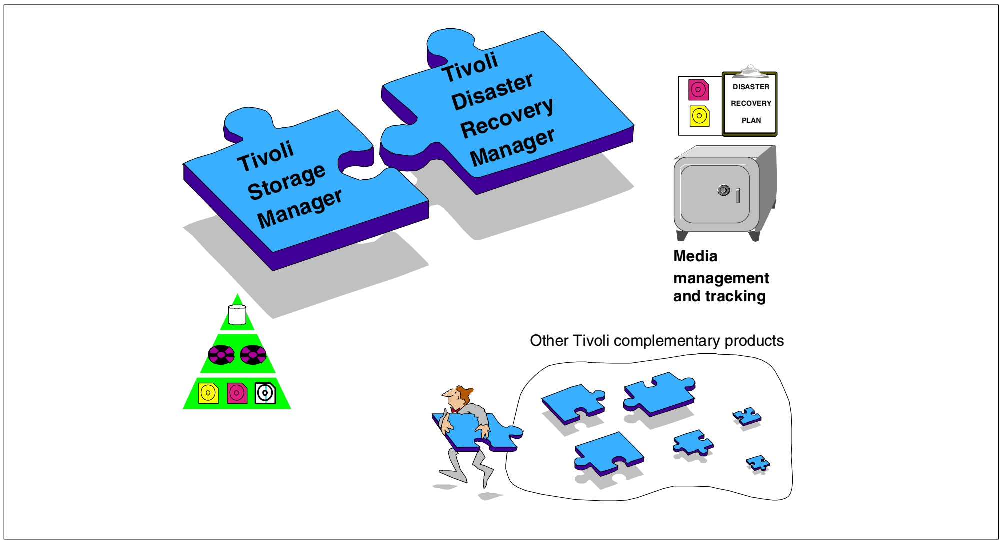


Figure 74. Tivoli Storage Manager and Tivoli Disaster Recovery Manager

One of the key features of Tivoli Storage Manager and Tivoli Disaster Recovery Manager is their ability to track media in all states that it could possibly be, such as onsite, in transit or in a vault. Because of the criticality of data in the production environment, you need to implement controls to make sure that all previously backed up data can be found and restored in a reasonable amount of time. Figure 75 shows a typical scenario using Tivoli Disaster Recovery Manager. There are two main data objects that Tivoli Storage Manager and Tivoli Disaster Recovery Manager will look after for you:

1. Tivoli Storage Manager server database backups: A database backup is the heart of Tivoli Storage Manager and therefore, is vital for server recovery.
2. Copy storage pool data: As Tivoli Storage Manager backs up the clients, new data is stored in the primary pools. For offsite storage requirements, you use copy storage pools. The BACKUP STORAGEPOOL command copies all new primary storage pool files to the copy storage pool. This ensures that the copy storage pool is an up-to-date reflection of your most recent backup. Each time the primary pool is backed up to the copy storage pool (and this is recommended to be performed daily), the newly generated tapes should be sent offsite.

In addition, Tivoli Disaster Recovery Manager allows you to document and prioritize your inventory of critical client nodes and their hardware and software recovery requirements.

Therefore, in a typical protected Tivoli Storage Manager environment, after each night’s normal client backup, the copy storage pools are also updated with the newly arrived data. Then a server database backup is done. The latest generated volumes are sent to a safe location and *recovery plan file* is regenerated by Tivoli Disaster Recovery Manager to make sure it includes the latest information. As data expires from the onsite pools, it also expires from the offsite pools, as well as unneeded database backups. Tivoli Disaster Recovery Manager also tracks such media as they become empty so that you can report on free tapes which can be brought back onsite for re-use.

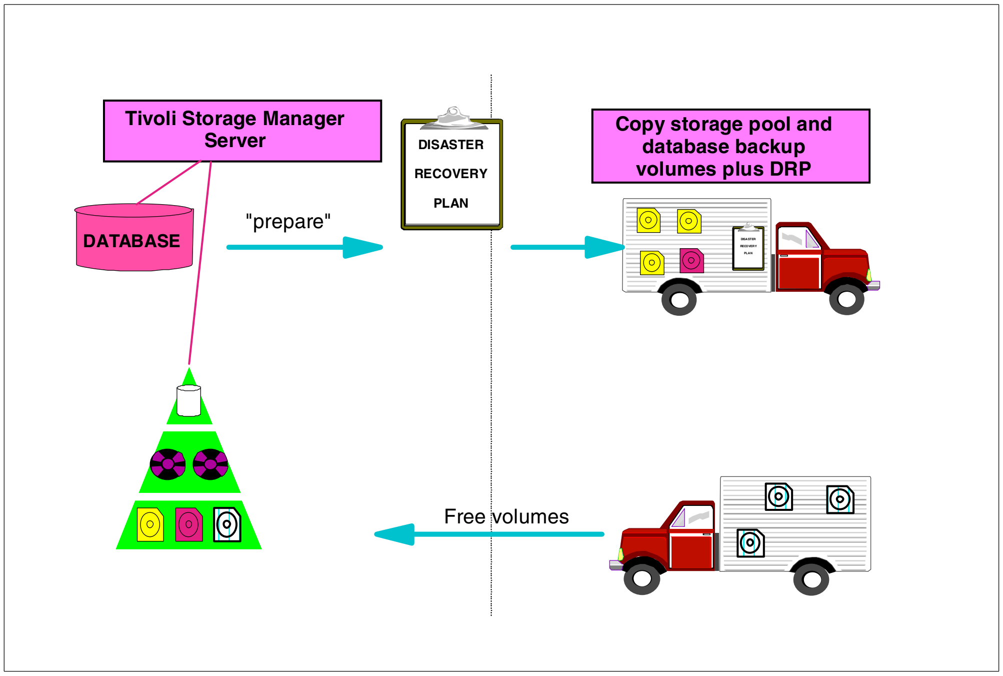

Tivoli Storage Manager Server
DATABASE
"prepare"
DISASTER RECOVERY PLAN
Copy storage pool and database backup volumes plus DRP
Free volumes

Figure 75. Disaster recovery flow

9.1.2 Disaster Recovery Manager volume tracking

Tivoli Disaster Recovery provides a number of levels of volume tracking. Although we refer to tape volumes here and in following sections, we can just as easily use optical volumes or even virtual volumes (discussed in Chapter 12, “Virtual Volumes” on page 145) for offsite data requirements. Tivoli DRM volume management includes:

- **Identifying what offsite volumes are needed for a given recovery:** Tivoli DRM knows the volumes which are associated with each primary storage pool so that you can initiate a complete recovery of all storage pools, or only a partial recovery, depending on the extent of the disaster. You can also configure DRM to track volumes only from certain storage pools. This might be useful in the case where you have some critical client nodes where you offer full offsite protection, and other less critical nodes where limited or no offsite protection is available. In this case, you would separate the two sets of clients to use different primary storage pools and only use Tivoli DRM for the critical pools.
- **Integrating with tape management systems:** Since Tivoli Disaster Recovery Manager is fully integrated with tape management, every time a new tape is created in the corresponding copy storage pools, it is automatically eligible for offsite movement.
- **Recycling partially filled volumes:** Offsite volumes are reclaimed just like onsite volumes. Tivoli DRM allows you to see which volumes have reached an empty state because of reclamation, so that you can request them to be returned onsite.
- **Tracking offsite volumes:** This is one of the strongest features of Tivoli Disaster Recovery Manager. Tivoli DRM manages tapes by assigning a special, predefined set of states to each offsite tape. Depending where the

tape should be, there are two possible directions for a tape: from Onsite to Offsite and from Offsite to Onsite. The first starts during normal backup processing to save up-to-date data to the copy storage pool. The tapes pass through a number of states in their journey from the production tape library to the safe vault. Then, time elapses while the tape remains offsite, ready to be used for restore in the event of disaster. During this time, data is gradually expiring from the tape. When the tape finally reaches its reclamation threshold, it is reclaimed by normal processes. Once empty, it moves in the reverse direction, that is, it is returned onsite for reuse. Figure 77 illustrates the tape states as explained below.

Onsite —> Offsite

Tapes are transitioned in the following order - *Mountable, NotMountable, Courier, Vault*.

1. **Mountable**: Newly created copy storage pool and database backup tapes are in a mountable state, so long as they are still online in the tape library. An onsite disaster would destroy these tapes also, since they have not yet been taken safely offsite. Therefore Tivoli DRM knows that these tapes are not part of the guaranteed recovery set when creating its recovery plan. The next step for these tapes is to remove them from the tape library, and, using DRM commands, change them to the NotMountable state.
2. **NotMountable**: This state means that the volumes are still in the onsite location but are not online to Tivoli Storage Manager. They could be in the datacenter itself, in a locked room near the tape library or in a different floor, waiting to be taken offsite. The actual physical location is not important for Tivoli Storage Manager. What is important is that volumes are still onsite and it is assumed they would also be destroyed in a disaster. This state only changes when they are taken offsite, usually by courier pickup. Again, a DRM command will change the state of the eligible volumes.
3. **Courier**: This state is an intermediate situation where you consider the tapes as being in transit. This means that you are still considering the physical transition from your onsite location to the vault as a potential risk. For example, suppose that you have a service agreement with an external company to carry out those tapes for you. Although you rely on the staff and the company to safely move all tapes, some (or all) tapes could be lost or damaged during transportation. This is a critical step for tape movement (and proper disaster recovery) because you cannot assume that a tape in the courier state has safely reached its destination. This is a protection that Tivoli Disaster Recovery Manager takes into account if you create a recovery plan file just after changing the tape state to courier. Once you are sure that all tapes have reached their destination, it is safe to change to the next and final state. Figure 76 illustrates a tape movement still in courier state.

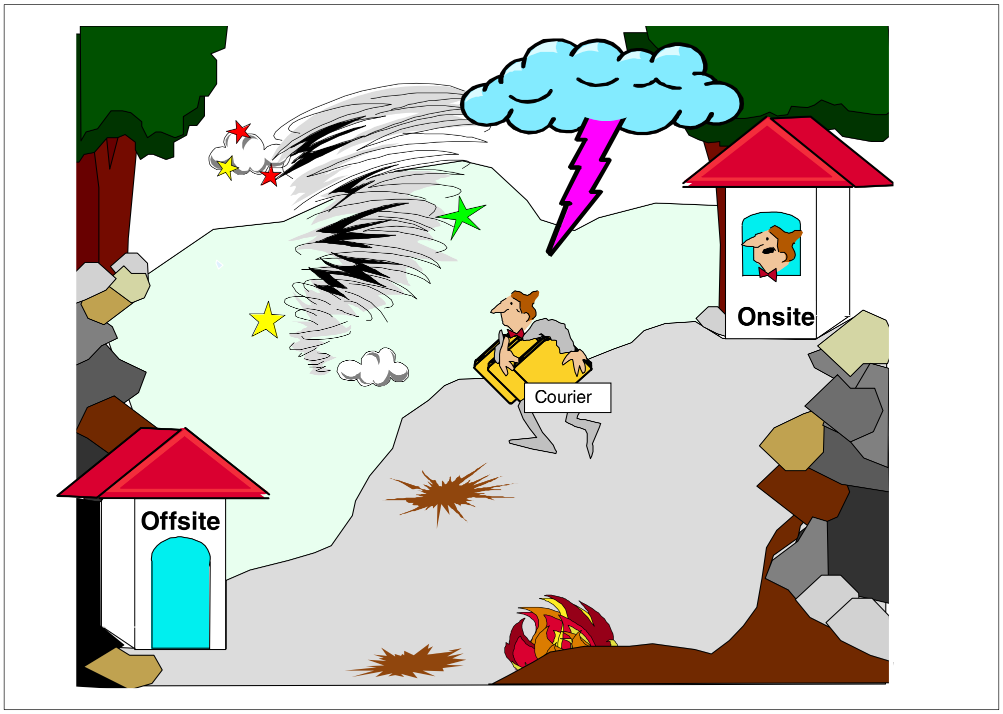


Figure 76. Tape movement and the Courier state

4. **Vault**: Once you have acknowledgement that the volumes have been safely received, you can change them to the Vault state, which is the final state to send a tape from an onsite location to offsite.

Once the volumes are offsite, they stay there until they are eventually reclaimed (or are required for an actual recovery). When they are empty, they are then ready to begin the reverse trip. Tivoli DRM query commands allow you to see which volumes can be retrieved so you can issue a list to your valut administrators.

Offsite —> Onsite

Tapes are returned onsite in the following order - *VaultRetrieve, CourierRetrieve, OnsiteRetrieve*.

1. **VaultRetrieve**: When all data in an offsite tape is no longer valid, the tape state is automatically changed to VaultRetrieve, meaning that it is available to be brought back onsite for usage. This happens for both Tivoli Storage Manager expired database backups and empty reclaimed copy storage pool tapes. Typically you would send this list to your storage vault administrators, for example once a week so they know which tapes to find and send back to you. Tapes that are in VaultRetrieve status are still part of the safe recovery set until they are physically removed from the vault.
2. **CourierRetrieve**: You change a tape to this state when you know that the tapes have been taken from the vault and are in transit back onsite. Similarly to the Courier state, these tapes may or may not be preserved safely in the event of a disaster, so it is important to know which they are. Once the volumes are correctly acknowledged on delivery onsite, they are ready to change to the next state.

3. **OnsiteRetrieve**: Once the volumes are back onsite, you can use them as scratch tapes. Usually they will be loaded back into the tape library.

Important:

Tivoli Disaster Recovery Manager allows you to skip some intermediate transition states. For example, you could choose to change your tapes directly from the Mountable to the Vault state. However, we do not recommend you do this, because it means there is less control over exactly which tapes are and are not safely available for recovery in the event of a disaster.

Figure 77 shows the complete lifecycle of a tape volume and the associated states.

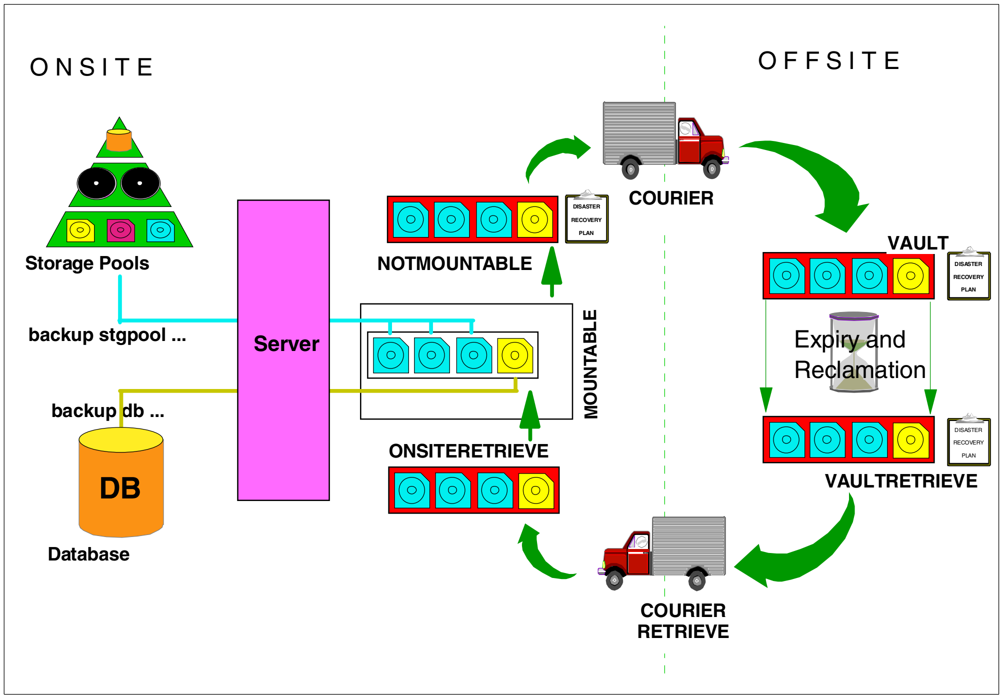


Figure 77. Offsite tape states

9.1.3 Focus on recovery

Facilitating data recovery is one of the most critical customer requirements in the area of storage management. Perhaps the most common recovery scenario is the restoration of user files. Another common scenario in the distributed world is restoration due to the loss of a disk drive. Recovery scenarios come in many varieties, and a complete solution should accommodate them all.

A system administrator should also be able to recover the storage management server inventory. They should be prepared for less-frequent catastrophic disasters, where the client and/or server environment is lost.

Figure 78 illustrates two recovery possibilities for data recovery. One is to recover the Tivoli Storage Manager server with the assistance of recovery plan files created by Tivoli Disaster Recovery Manager, so that you can recover all your vital data. The other possibility is to use Tivoli Storage Manager backupset features to recover directly from locally attached tape drives. This situation does not necessarily require the DRM recovery plan file because of backup sets can be used independently of the Tivoli Storage Manager server. Both scenarios are available could also be used simultaneously, depending solely on your backup/recovery strategy and criticality.

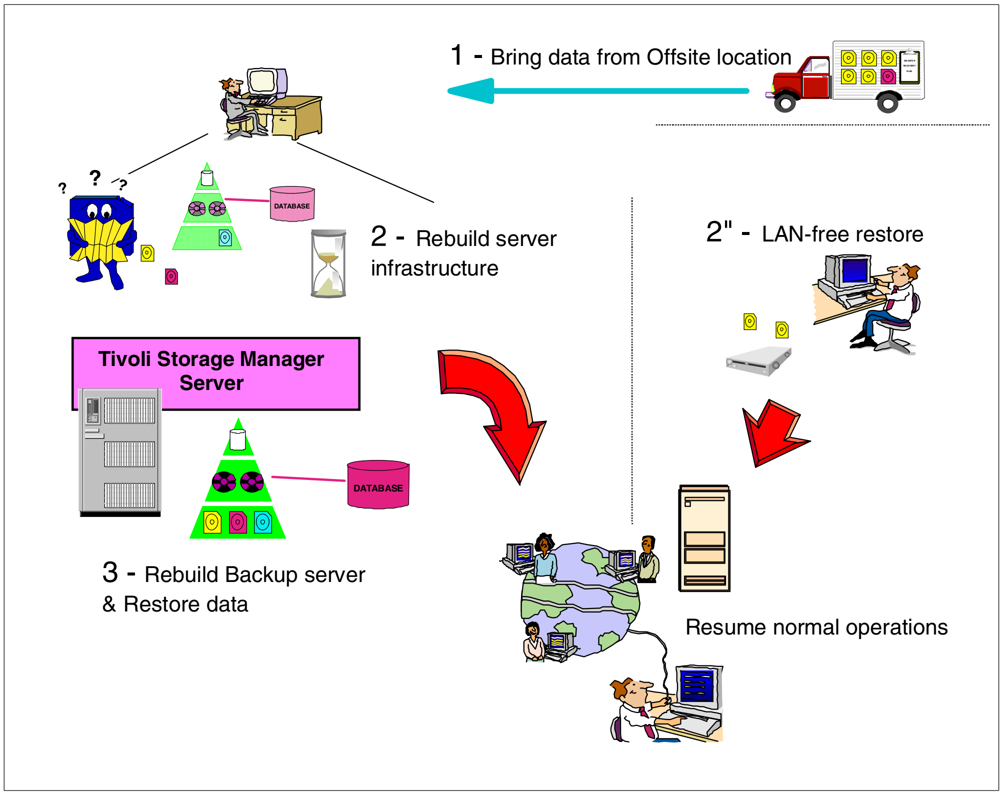

1 - Bring data from Offsite location
? ? ?
DATABASE
2 - Rebuild server infrastructure
2" - LAN-free restore
Tivoli Storage Manager Server
DATABASE
3 - Rebuild Backup server & Restore data
Resume normal operations

Figure 78. Recovery order and possibilities

You should also consider a minor disaster, where only a small portion of your data has been lost, but you can still use the same Production hardware (machine, tapes and so forth). Tivoli Storage Manager supports recovery to the state of the last backup for files, directories, logical volumes, disks, or an entire system image. This includes file and file system attributes. Optionally, a system administrator should be able to recover to the time of any of the last several backups. Another function would be restoration of a file only if the version stored on the server is a more recent backup version than that already existing on the client. This "restore if newer" feature could be run in a situation where the extent of file loss or corruption is unknown.

Figure 79 illustrates an onsite file restore processing using the backup tapes available.

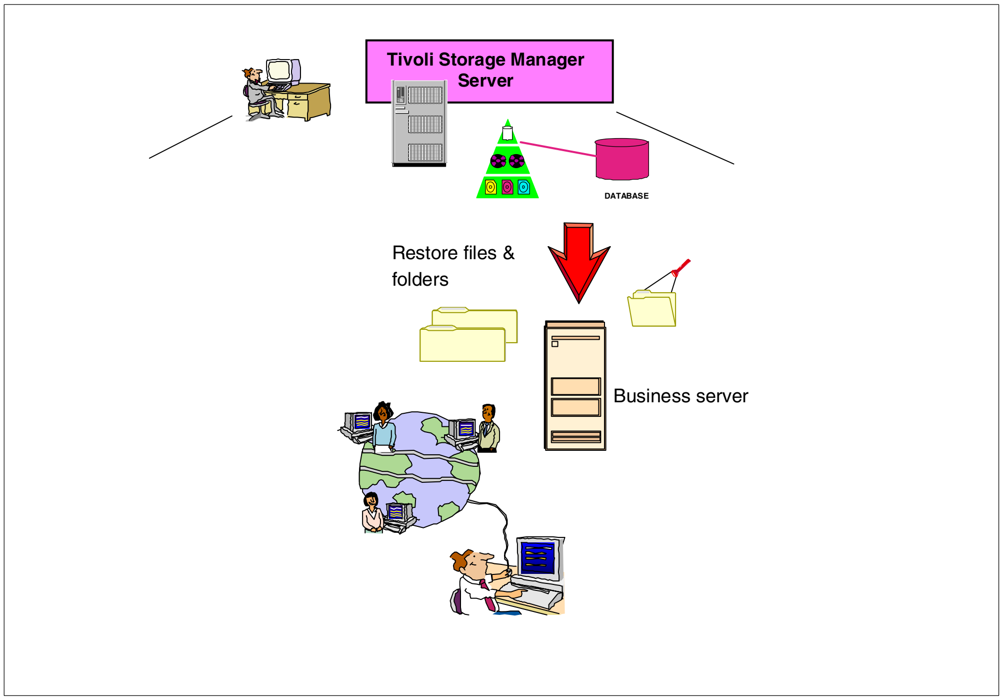


Figure 79. Restoring single files

9.1.4 Speed to recovery

One of the most important requirements for disaster recovery is to minimize the time to recovery. This is done through planning, preparing and testing for recovery, not just for backup. A complete recovery solution should not stop once backups are complete; but instead should assist in preparing a disaster recovery plan and then help recover the data when needed. It should also do everything possible to automate and facilitate the recovery.

Since data recovery scenarios are often unique, part of speeding recovery is to have flexibility in how restores are done. For example, authorized administrators should be allowed to restore on behalf of clients in the event an owner of shared data is unavailable. It is not always possible to recover all lost data immediately. Administrators should also be able to schedule restores to allow highest priority data to be restored first. This is why the preliminary steps, including the Business Impact Analysis, are so important, in order to prioritize the recovery requirements.

Some applications are critical enough to warrant hot site recovery for the storage management server. This requires a second storage management server at a remote site that is ready to go on-line immediately upon failure of the primary storage management server. The secondary server might not be a true mirror of the primary server due to network delays in server updates. A hot site recovery requires a higher investment in hardware and software, but can be viable for certain business applications.

9.2 What should be considered a disaster?

A disaster is a catastrophic interruption of business processing that destroys the Tivoli Storage Manager server, clients, or both.

There are many levels to consider for a disaster. You can imagine a situation where a user deletes a file and requests a restore (to them it might indeed be ’a disaster’). A complete directory might be lost; an application server might be infected with a virus, causing an unknown extent of damage; a disk might be lost on a fileserver; or a full datacenter building with many machines might be completely destroyed. Depending on the business requirement, you will need to answer bigger questions, like what kind of recovery do you need and how long do you have to provide it. Figure 80 shows some of the well-known techniques that you can apply to your environment to safeguard from data loss.

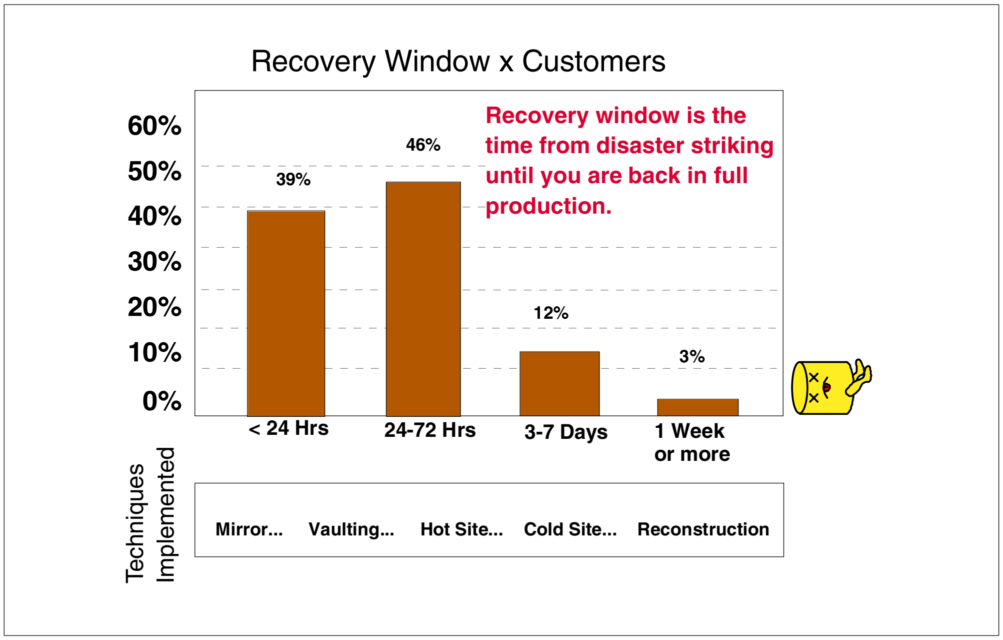


Figure 80. Disaster recovery techniques

Tivoli Storage Manager and Tivoli Disaster Recovery Manager can provide assistance to you in the following situations:

- **Reconstruction**: A reconstruction means that you want to rebuild your previous working environment at the same place and to the same state as it was before. This also means that you do not have all required machines available to start and therefore, you will need to contact local dealers to provide you all the hardware and software needed. Assuming that you have saved your vital data to an alternate offsite location, Tivoli Disaster Recovery Manager will help you rebuild your Tivoli Storage Manager server, so that you can then restore the client nodes up to the last known saved position.
- **Cold Site**: In this scenario, you may have all spare hardware and software available, but not properly configured or ready for usage. Some preliminary steps are still required to prepare the infrastructure. In this case, Tivoli Storage Manager will have to be installed with all additional required software. You then restore your latest database backup, and restore the storage pools, using the DRM recovery plan. You can then restore the clients, starting with the most critical.
- **Hot Site**: Generally, a hot site will have all the infrastructure hardware and software in place, so that you need to focus on business information

recovery, rather than infrastructure. However, you will still need to bring your data to the hot site location. Assuming that your hot site has all basic software installed (Tivoli Storage Manager included), you will need to recover your backup server and application data by using the recovery plan file and database backups.

- **Vaulting**: This provides you all hardware, software and data requirements for recovery. In this case, you need to concentrate solely on business recovery, since all data is already available to you. Tivoli Disaster Recovery Manager offers a complete set of tape state controls to make tape vaulting easy and precise. All offsite tapes can have one of the following stages: Mountable, NotMountable, Courier, CourierRetrieve, Vault, VaultRetrieve, OnsiteRetrieve. This governs how the recovery will be affected if, for example, the volumes are in transit to offsite (courier) and did not reach the offsite location (vault).
- **Mirror**: A mirror site normally replicates all required data simultaneously, so that whenever you have a failure in the primary image, the copy (or copies) can be used without interruption of business activity. This is the only situation that you may not need Tivoli Disaster Recovery Manager at all, since you already have an alternate, secure and up-to-date condition for your data. However, you could still consider using Tivoli Disaster Recovery Manager to control other machines that are not being mirrored to this site.

Backup operations can guarantee the first level of protection, which is when main data is somehow lost, inconsistent or damaged. Unfortunately, if your backup tapes are located in the same place as your production data, then you risk losing your only safeguard. Figure 81 shows a typical situation where offsite data storage is used. The latest file changes are sent to the offsite location, so that you can use it together with previous data sent.

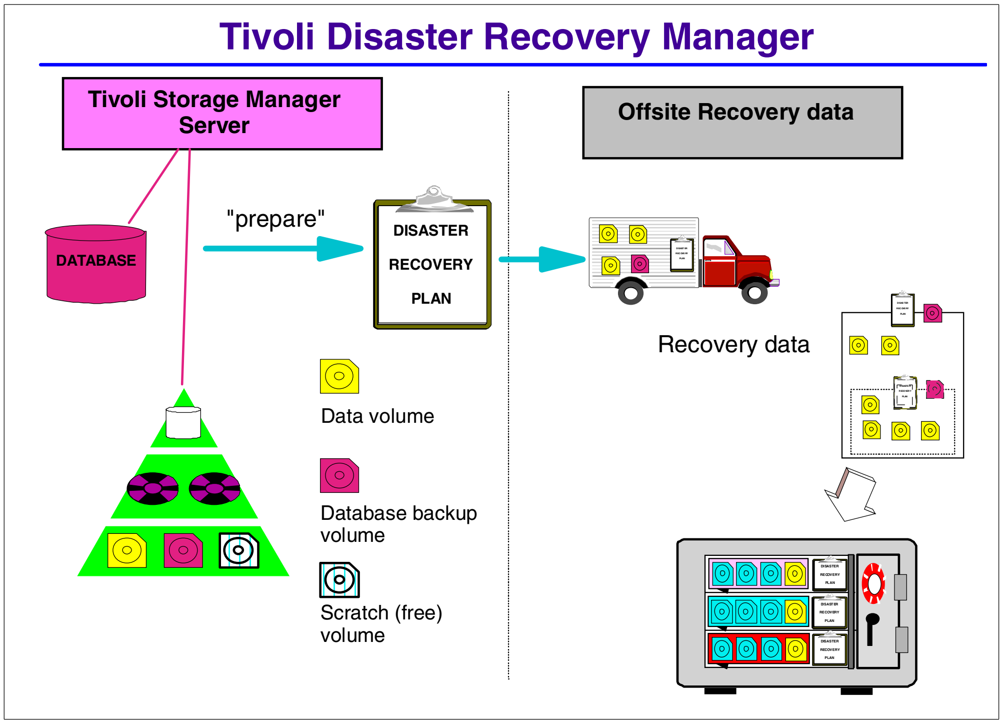


Figure 81. Sending data to offsite location

Backup data is located offsite to protect it from damage. Data does not need to be lost or destroyed in order to be unavailable. In recent disasters, data inaccessibility appeared to be caused as much by condemned buildings and evacuations as by destruction. Companies need to plan for the physical accessibility of data, not just the survival of data.

In today's world, however, it all depends on who you are, where you are, and what you were doing. Today's disaster can be characterized by the business you are in (for example, banking, e-commerce, distribution, or manufacturing). In banking or e-commerce, time is very important. One hour of downtime—regardless of how it is caused—can mean business loss amounting to millions of dollars. In the e-commerce world, if you cannot satisfy your customer's requests, they may go to a competitor and never return to your site. The potential losses may not be as large in manufacturing or distribution. But whatever the business is, the longer the downtime, the greater the loss that the business incurs and the greater the risk that the business may never recover from the loss.

Disaster recovery management is accomplished with Tivoli Storage Manager by a combination of:

- Backing up client data to the Tivoli Storage Manager server
- Backing up the server database to removable media and storing the media offsite
- Backing up the primary storage pools and storing the media offsite
- Using the disaster recovery plan file to assist with the Tivoli Storage Manager server recovery
- Optionally using LAN-free recovery options, such as backupsets where available and appropriate to improve recovery

- Optionally using virtual volumes to save data, recovery plan file and database information electronically to an alternate Tivoli Storage Manager server

9.3 Recovery strategy for the Tivoli Storage Manager server

Tivoli Disaster Recovery Manager simplifies the disaster recovery planning process for the Tivoli Storage Manager server by generating a recovery plan file that is based on a pre-defined recovery strategy. The recovery plan file contains the information and procedures necessary to help restore the key components of the Tivoli Storage Manager server. The content of the plan file includes:

- Installation-specific server recovery instructions
- List of Tivoli Storage Manager database backup and copy storage pool volumes required to perform the recovery including the offsite location where the volumes reside
- Devices required to read the database backup and copy storage pool volumes
- Space requirements for the Tivoli Storage Manager database and recovery log
- Copy of Tivoli Storage Manager configuration files
- Shell scripts and Tivoli Storage Manager macros for performing server database recovery and primary storage pool recovery

9.3.1 Creation of up-to-date disaster recovery plan

To make the creation and maintenance of the server disaster recovery plan easier, the `prepare` command automatically queries the required information from the Tivoli Storage Manager server and creates the recovery plan file. The `prepare` command can be scheduled using the Tivoli Storage Manager central scheduling capabilities.

9.3.1.1 Auditable plan for Tivoli Storage Manager server

The recovery plan file contains the information and procedures necessary to assist with the recovery of the Tivoli Storage Manager server. The information in the plan file includes:

- Site-specific server recovery instructions and information as defined by the administrator (for example, contact names and telephone numbers for critical people and their backups)
- The sequence of steps necessary to recover a Tivoli Storage Manager server
- List of Tivoli Storage Manager database backup and copy storage pool volumes required to perform the recovery (the offsite location where the volumes reside is included)
- Devices required to read the database backup and copy storage pool volumes
- Space requirements for the Tivoli Storage Manager database and recovery log
- Copy of Tivoli Storage Manager server options file, device configuration file, and volume history information file
- Shell scripts (on UNIX), REXX scripts and JCL (on MVS), and Tivoli Storage Manager macros for performing server database recovery and primary storage pool recovery

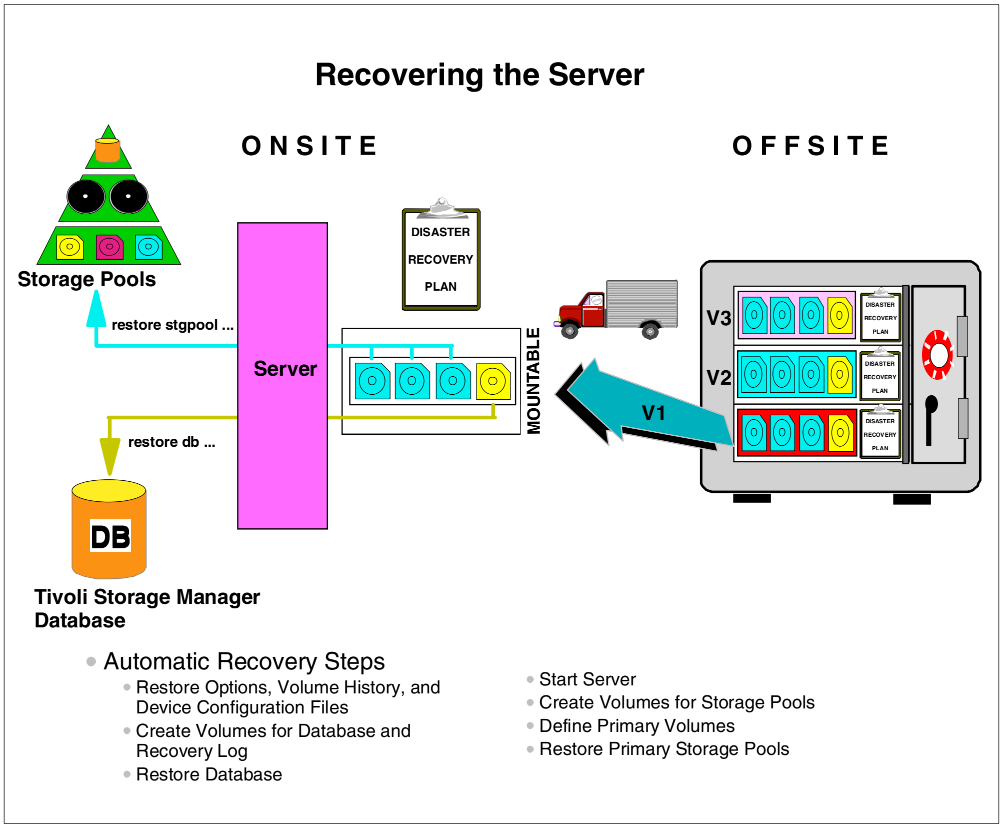
9.3.1.2 Offsite recovery media management

Knowing the location of offsite recovery media is critical to the successful implementation of a disaster recovery management plan. The offsite recovery media management function provides:

- Determination of what database and copy storage pool volumes need to be moved offsite and back onsite
- Automatic ejection of volumes from an automated library
- Tracking of the media location and state in the Tivoli Storage Manager database

This function allows database backup volumes and copy storage pool volumes to be treated as logical collections that are selected to move offsite for safekeeping and onsite for use. The reclamation of offsite volumes includes the capability to specify the number of days to retain a Tivoli Storage Manager database backup series. After the expiration interval is reached, the data on the media is no longer considered to be valid. The media can then be reused.

Figure 82 illustrates how your offsite data cold be used to recover your environment. Note that "V1" is the point in time requested, and therefore, you can not only rebuild the latest one, but also any specific point in time data that you still have saved. The execution of the recovery scripts (which perform the Automatic Recovery Steps in the figure) starts from after you have re-installed the operating system and Tivoli Storage Manager server code on your replacement server hardware.



- Automatic Recovery Steps
  - Restore Options, Volume History, and Device Configuration Files
  - Create Volumes for Database and Recovery Log
  - Restore Database
  - Start Server
  - Create Volumes for Storage Pools
  - Define Primary Volumes
  - Restore Primary Storage Pools

Figure 82. Restoring a Tivoli Storage Manager server

9.3.1.3 Storage of client recovery information

Tivoli Disaster Recovery Manager allows the machine information needed to help recover the Tivoli Storage Manager clients to be stored in the Tivoli Storage Manager database. This information includes:

- Tivoli Storage Manager client machine location, machine characteristics, and recovery instructions
- Business priorities associated with the Tivoli Storage Manager client machines
- Description, location, and volume/diskette labels of Tivoli Storage Manager client boot media

9.3.1.4 Centralized management of the disaster recovery process

Based on the server information contained in the recovery plan file and the client information stored in the Tivoli Storage Manager database, a centrally managed, pre-defined disaster recovery management strategy can be implemented. Tivoli Disaster Recovery Manager assists customers with the recovery of the Tivoli Storage Manager server by providing:

- Offsite media requirements—This includes the name and location of the server database and copy storage pool volumes.
- Sequence of steps to recover the Tivoli Storage Manager server machine—The recovery plan file contains the sequence of steps necessary to recover an Tivoli Storage Manager server.
- Automated command generation—The `prepare` command automatically generates and saves commands for performing server database recovery and primary storage pool recovery.

Tivoli Disaster Recovery Manager also assists customers with recovery of Tivoli Storage Manager client machines:

- Performing damage analysis—Tivoli Disaster Recovery Manager provides query capabilities to determine what client machines were lost in the disaster and need to be recovered.
- Reporting recovery requirements for client machines—Based on the information acquired in the damage analysis step, Tivoli Disaster Recovery Manager reports machine requirements and offsite boot media requirements.
- Recovery order based on business priorities—Tivoli Disaster Recovery Manager allows priorities to be associated with client machine definitions. This information is used to determine the order to recover machines.

9.3.2 Additional DR issues

Disaster recovery goes far beyond simply technical measures. To have a fully operational and prepared environment, you must also pay attention to additional issues, as highlighted.

9.3.2.1 Hardware system requirements

This is one of the things you have to consider for recovery. Tivoli Disaster Recovery Manager will create a recovery plan file based on the information and space allocation that you have for the Tivoli Storage Manager production server machine. This means that you must evaluate if you are going to have a similar machine for offsite recovery and make the changes to fit the new environment.

9.3.2.2 Additional Operating System recovery steps

Depending on the operating system you have Tivoli Storage Manager installed, you may need to send special compact disk (CD) or tape images (for the specific OS recovery steps) to the offsite location. For example, if you have an AIX machine, this would be fully supported by the `mksysb` operating system command, so that you would have a valid, bootable tape image of your present configuration.

9.3.2.3 Recovery testing

One of the challenges that you must consider when designing a recovery solution is to make sure that it is tested. A good approach is to create all documents, operating system tapes, special hardware requirements, installations scripts and send them to the offsite location as a "Disaster Recovery starter kit". Then, once a year you perform a complete recovery test to make sure the documents are accurate for recovery. You should then incorporate any changes which were uncovered during your test.

Chapter 10. Tivoli Space Manager

The most recent version of this material is maintained on the ITSO Internet Web pages at:

```
http://www.redbooks.ibm.com/solutions/systemsmanagement/tsm/c_hsmcli.html
```

This chapter discusses the complementary Tivoli Storage Manager product, Tivoli Space Manager (formerly Hierarchical Storage Manager or HSM client). It introduces the concept of Tivoli Space Manager and discusses various features of the product.

10.1 Introduction

Disk subsystems are becoming less expensive every day. The cost per byte of data stored on a disk has dropped dramatically from just a few years ago. At the same time, both the amount of data being stored on disk and the cost to manage and administer this stored data is increasing dramatically. Even with the reduction in hardware costs, the increase in the amount of data is taxing the ability of file servers to hold and manage this data.

File owners have been and continue to be disinterested in actively managing their storage requirements, due to factors such as lack of knowledge, more pressing priorities, and a lack of tools. They want to be able to find their data quickly and access it without the hassles of swapping disks and mounting tapes. These attitudes lead to a situation where significant amounts of data are stored on disk, not because data needs to be there or is actively being used, but because it is convenient to do so and no-one is concerned with removing old or unneeded data.

Over twenty years ago, mainframe computer sites were facing a similar situation: explosive data growth and the requirement to have fast access to less current data, but with extremely expensive disk devices. The industry came up with the concept of having the computer manage the data storage by moving less frequently accessed data to lower cost storage media while presenting to the end user the impression the data was still on disk. This became known as *hierarchical storage management* (or HSM).

Tivoli Space Manager maximizes usage of existing storage resources by transparently migrating data off workstation and file server hard drives based on size and age criteria, leaving only a stub file. If and when the migrated data is accessed, Tivoli Space Manager transparently migrates data back onto the local disk. In doing so, Tivoli Space Manager relieves the user from the task of manual deleting and archiving of data on his workstation.

Tivoli Space Manager is a complementary product which is available on all Tivoli Storage Manager servers. The Space Manager (HSM) client is currently available on AIX and Sun Solaris operating systems. Therefore, the supported HSM clients can send data to any Tivoli Storage Manager server which also has installed the Tivoli Space Manager. Tivoli Storage Manager is a pre-requisite to implementing Tivoli Space Manager.

Tivoli Space Manager provides the basic functionality of space management by automatically migrating data based on file size, number of days since it was last accessed, or a combination of both. Once migrated, Tivoli Space Manager automatically recalls a file if it is accessed and restores it to its original location in the file system.

Tivoli Space Manager maintains data integrity and security of data by working closely with the operating system. It provides a graphical user interface and commands that can be used to display information about files, including whether they have been migrated.

10.2 HSM migration

Files are migrated by Tivoli Space Manager from the original file system to storage devices connected to a Tivoli Storage Manager server. Each file is copied to the server and a stub file is placed in the original file's location. Using the facilities of storage management on the server, the file is placed on various storage devices such as disk and tape.

Tivoli Space Manager migrates only regular files on locally mounted file systems. It does not migrate character special files, block special files, FIFO special files (named pipe files) or directories.

Note

Be careful not to confuse migration in the HSM sense with storage pool migration. Storage pool migration is the process where client data (which could be backup, archive or HSM data) moves through the Tivoli Storage Manager storage hierarchy, typically from disk to tape or optical. Storage pool migration happens entirely within the Tivoli Storage Manager server. HSM migration is the process of moving data from an HSM client to the Tivoli Storage Manager where it will be stored in a storage pool. So, once a file has been migrated (HSM) from a client to the server, it could subsequently be migrated (server) to another server storage pool.

There are two types of HSM migration: automatic and selective.

10.2.1 Automatic

With automatic migration, Tivoli Space Manager monitors the amount of free space on your file systems. When it notices free space shortage, it migrates files off the local file system to the Tivoli Storage Manager server storage based on the space management options that have been chosen. Tivoli Space Manager monitors free space in two ways: threshold and demand.

10.2.1.1 Threshold

Threshold migration maintains your local file systems at a set level of free space. At an interval specified in the options file, Tivoli Space Manager checks the file system space usage. If the space usage exceeds the high threshold, files are migrated to the server by moving the least recently used files first. When the file system space usage reaches the set low threshold, migration stops. Threshold migration can also be started manually.

10.2.1.2 Demand

Tivoli Space Manager checks for an out-of-space condition on a file system every two seconds. If this condition is encountered, Tivoli Space Manager automatically starts migrating files until the low threshold is reached. As space is freed up, the process causing the out of space condition continues to run. You do not receive out-of-space error messages while this is happening.

10.2.2 Selective

You can tell Tivoli Space Manager to selectively migrate a file immediately to the server's storage. As long as the file meets the space management options, it will be migrated. The file does not need to meet age criteria, nor does the file system need to meet space threshold criteria.

10.2.3 Pre-migration

Migration can take a long time to free up significant amounts of space on the local file system. Files need to be selected and copied to the Tivoli Storage Manager server, which may involve tape mount, and a stub file must be created in place of the original file. To speed up the migration process, Tivoli Space Manager can be told to implement a pre-migration policy.

After threshold or demand migration completes, Tivoli Space Manager continues to copy files from the local file system until the pre-migration percentage is reached. These copied files are not replaced with the stub file, but they are marked as pre-migrated.

The next time migration starts, the pre-migrated files are chosen as the first candidates to migrate. If the file has not changed since it was copied, the file is marked as migrated and the stub file created in its place in the original file system. No copying of the file needs to happen, as the server already has a copy. In this manner, migration can free up space very quickly.

10.3 Recall

Recall is the process for bringing back a migrated file from Tivoli Storage Manager to its original place on the local file system. A recall can be either transparent or selective.

10.3.1 Transparent

From a user or running process perspective, all the files in the local file system are actually available. Directory listings and other commands which do not require access to the entire file will appear exactly as they would if the HSM client was not installed. When a migrated file is needed by an application or command, the operating system initiates a transparent recall for the file to the Tivoli Storage Manager server. The process is temporarily halted while the file is automatically copied from the server's storage to the original file system location. Once the recall is complete, the halted process continues without requiring any user intervention. In fact, depending on how long it takes to recall the file, the user may not even be aware that HSM is used.

After a recall, the file contents are on both the original file system and on the server storage. This allows Tivoli space Manager to mark the file as pre-migrated and eligible for migration unless the file is changed.

10.3.2 Selective

Transparent recall only recalls files automatically as they are accessed. If you or a process need to access a number of files, it may be more efficient to manually recall them prior to actually using them. This is done using selective recall.

Tivoli Space Manager batches the recalled file list based on where the files are stored. It recalls the files stored on disk first, then recalls the files stored on sequential storage devices such as tape.

10.4 Advanced transparent recall

Advanced transparent recall is available only on AIX platforms. There are three recall modes: normal, which recalls a migrated file to its original file system; migrate-on-close; and read-without-recall.

10.4.1 Migrate-on-close

When Tivoli Space Manager uses the migrate-on-close mode for recall, it copies the migrated file to the original file system, where it remains until the file is closed. When the file is closed and if it has not been modified, Tivoli Space Manager replaces the file with a stub and marks the file as migrated (since a copy of the file already exists on the server storage).

10.4.2 Read-without-recall

When Tivoli Space Manager uses read-without-recall mode, it does not copy the file back to the originating file system, but passes the data directly to the requesting process from the recall. This can only happen when the processes that access the file do not modify the file, or if the file is executable, the process does not execute the file. The file does not use any space on the original file system and remains migrated (unless the file is changed; then Tivoli Space Manager performs a normal recall).

10.5 Reconciliation

Tivoli Space Manager uses reconciliation to maintain synchronization of the local file system and Tivoli Storage Manager. Reconciliation builds a migration candidates list.

Reconciliation can be started manually or be allowed to happen automatically at intervals set in the options file and prior to threshold migration if the migration candidate list is empty.

10.5.1 Synchronization

Synchronization involves maintaining the Tivoli Space Manager database in sync with the actual files on the original file system. It ensures that for every stub file there is a valid file copy kept; for every original file on the original file system, there are no database entries; for pre-migrated files there is an entry in the Tivoli Space Manager database; and it updates status fields in the database.

For example, if you recall a file, change it, and immediately migrate it, Tivoli Space Manager has two copies of the file in its storage: the most recent one is valid; and there is also an obsolete one. Reconciliation will remove this obsolete file after its expiration interval has passed.

10.5.2 Building a new migration candidates list

Tivoli Space Manager uses the reconciliation process to build a prioritized list of files on the original file system that are eligible for automatic migration. The list is created based upon management class criteria and minimum file size. It is ordered according to the number of days since the file was last used, the file size, and the migration factors set in the options file. During threshold and demand migration, the list is used to select files to migrate in prioritized order. As the file is selected, it is checked again to ensure that it still meets the migration criteria.

A new migration candidate list is created each time reconciliation runs. The list can also be created at any time.

10.6 Options

Options to control Tivoli Space Manager are set in the client options file. These options set items such as what Tivoli Storage Manager server to use for Tivoli Space Manager functions, space management options, migration options, excluded file lists, and assigning management classes to files.

10.7 Backup/restore

You should not consider use of HSM and Tivoli Space Manager as a replacement for backup. It should be viewed as a form of space extension of local disk storage. When a file is migrated to the HSM server, there is still only one copy of the file available, since the original is deleted on the client and replaced by the stub. Also, Tivoli Space Manager maintains only the last copy of the file, giving no opportunity to store multiple versions. Therefore, the Tivoli Storage Manager backup-archive client must be used for files backup or archive before or after the file has been migrated by Tivoli Space Manager. You can specify that a file is not eligible for HSM migration unless a backup has been made first with the backup-archive client. If the file is migrated and the save Tivoli Storage Manager server destination is used for both backup and HSM, the server can copy the file from the migration storage pool to the backup destination without recalling the file.

Both files and stub files can be restored from a Tivoli Storage Manager backup. If you restore the entire file, it will become a normal resident file on the client and the migrated copy will be deleted from the HSM pool during the next reconciliation. If you do not want to restore the actual file data, you can use options on the HSM client to restore just the stub file without re-creating the file contents. In this case, the file will still remain in its migrated state.

10.8 Archive/retrieve

The Tivoli Storage Manager backup-archive client allows you to archive and retrieve copies of migrated files without performing a recall for the file first, providing the save Tivoli Storage Manager server is used for both HSM and backup-archive. The file will simply be copied from the HSM storage pool to the archive destination pool.

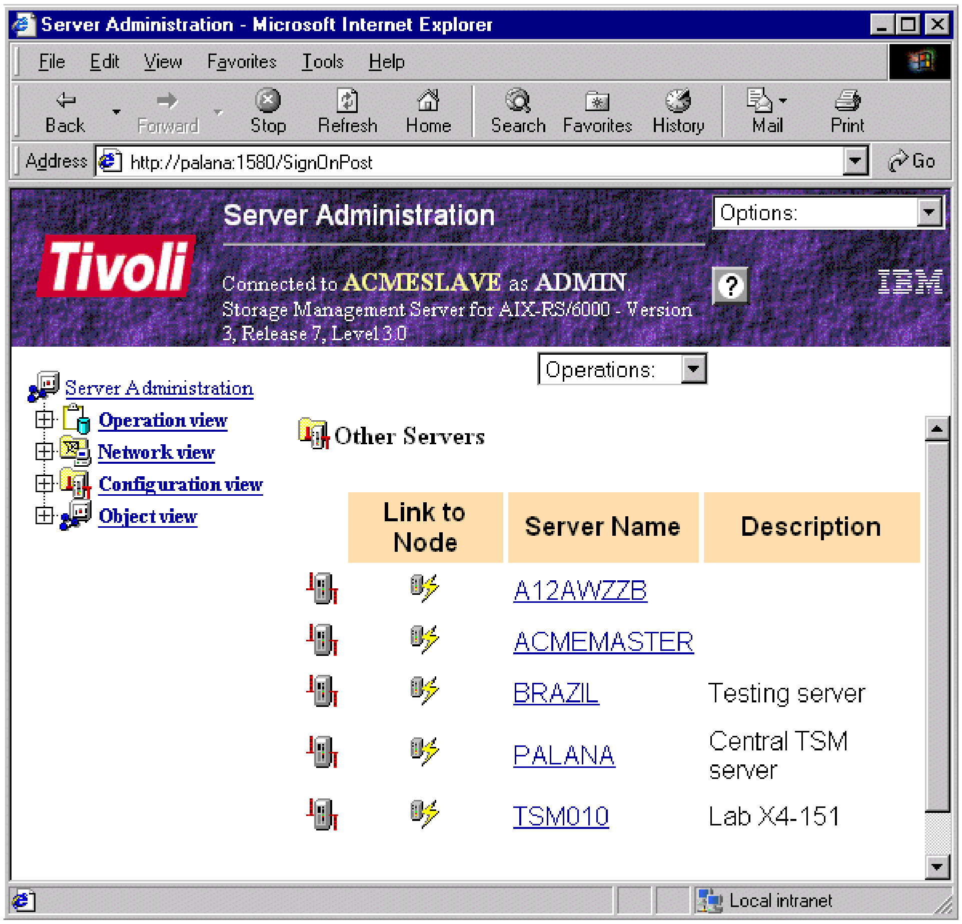
Chapter 11. Enterprise administration

The enterprise management capabilities in Tivoli Storage Manager includes comprehensive central management of multiple Tivoli Storage Manager servers, lights-out server automation and remote help desk support. These capabilities are designed to enhance ease-of-use, centralized but flexible control, automation and consistency, while meeting your requirements for protecting ever-increasing amounts of data in the most cost-effective manner.

These features enable you to easily and cost-effectively broaden your deployment of Tivoli Storage Manager servers to meet growing data availability requirements. Enterprise administration is available on all Tivoli Storage Manager Version 3 and above servers, including AS/400 systems and VM/ESA. This is now a standard no-charge feature in the base Tivoli Storage Manager product.

11.1 Enterprise management features

With the Enterprise Administration feature, the Tivoli Storage Manager Enterprise Console provides the administrator with several global views of their Tivoli Storage Manager deployment, as shown in Figure 83.



Connected to ACMESLAVE as ADMIN.
Storage Management Server for AIX-RS/6000 - Version 3, Release 7, Level 3.0

Link to Node	Server Name	Description
	A12AWZZB	
	ACMEMASTER	
	BRAZIL	Testing server
	PALANA	Central TSM server
	TSM010	Lab X4-151

Figure 83. Tivoli Storage Manager Enterprise Console

The enterprise management features include:

- **Enterprise console**

  A key element of the enterprise management is the user-friendly Tivoli Storage Manager enterprise console. This administrative web browser interface extends the capabilities of the current storage administrator command line interface. It enables the administrator to manage many Tivoli Storage Manager servers and clients from any Web-enabled platform in the enterprise. The Tivoli Storage Manager enterprise console is the integration point for all Tivoli Storage Manager functions and commands. It combines the Web browser interface with the backup-archive client web browser interface.

  The enterprise console provides centralized policy management and a global view of Tivoli Storage Manager deployment.

  Using the Tivoli Storage Manager enterprise console, the administrator can navigate, logon, and perform functions on any Tivoli Storage Manager server and Tivoli Storage Manager web backup-archive client from a supported Web browser.

  Some of the specific functions the administrator can perform from the enterprise console include:

  - Regular administration functions: All administration functions can be performed from the Enterprise Console.
  - Unified login for all functions: Once the administrator has signed onto a Tivoli Storage Manager Enterprise Console, the credentials used for login are transferable to all associated Tivoli Storage Manager servers and clients to which there is legitimate access. Multiple IDs and passwords do not need to be redefined or remembered. Clients have an option that allows them to restrict access based on administrator authority.
  - Encrypted password information: Secure Socket Layer (SSL) can be selected to encrypt password information between Tivoli Storage Manager servers and the Enterprise Console
  - Client operations through the web backup-archive clients: Through the web backup-archive clients, authorized administrators and help desk personnel can perform backup and restore functions on a Tivoli Storage Manager client remotely through a web browser interface. Security features implemented with unified logon enables help desk or administrators to recover data without having direct access to the machine while ensuring data security.

- **Web backup-archive clients**

  In addition to providing authorized administrator and help desk personnel with access to client functions, end users can also remotely access client data through this web browser interface. The Enterprise Console gives easy launch access for all configured web clients.

- **Enterprise monitoring**

  All aspects of server operation, such as server database utilization, server recovery log utilization, percentage of scratch tapes, off-line drives and license compliance across all the servers can be monitored.

- **Enterprise reporting**

  Enterprise reporting mechanisms provide the ability for the Tivoli Storage Manager servers to route their server and client events to another server. So, one server could receive the event messages from all the servers in the network, allowing consolidated reporting and querying.

- **Server scripts**

  Server scripts automate Tivoli Storage Manager server management operations by storing a sequence of server commands in the Tivoli Storage Manager database as a script with an assigned name. An administrator can run the script from the command line and from the Tivoli Storage Manager Enterprise Console, or schedule its execution using the administrative command scheduler in the Tivoli Storage Manager server. Server scripts can include Tivoli Storage Manager server commands, SQL statements, conditional logic (including return code checking) and variable substitution. Variable substitution allows scripts to become dynamic and reusable. A number of pre-defined server scripts are provided for loading into the Tivoli Storage Manager server database. Here are some examples from the supplied server scripts:

  - List nodes that have not accessed Tivoli Storage Manager for a specified number of days
  - Display nodes that have backup data exceeding a specified number of MB
  - Display the percentage of scratch volumes in a specified library

  The Tivoli Storage Manager Enterprise Console also enables administrators to create additional reporting scripts, modify existing ones and execute the scripts on any server in the environment.

- **Additional policy constructs**

  Additional policy constructs allow the automatic extension of the server database and recovery log when administrator-defined thresholds are reached. Triggers can be established to extend the database or recovery log when a specified capacity percentage is reached. The amount of space to be added can be specified as a percentage of the existing database or recovery log size.

- **Simplified volume formatting**

  Disk storage allocation for database, recovery log and storage pools is simplified through integration of the space format utility, DSMFMT, with the `define volume` command. This allows simple one step formatting and allocation of these types of volumes.

11.2 Enterprise administration features

The Enterprise Administration feature enables administrators to centrally manage multiple Tivoli Storage Manager servers through a single Enterprise Console.

The Tivoli Storage Manager Enterprise Administration provides the following capabilities:

- **Enterprise configuration and policy management**

  Allows Tivoli Storage Manager policy and configuration information to be defined a single time and then propagated to any number of managed Tivoli Storage Manager servers. A Tivoli Storage Manager server option defines a Tivoli Storage Manager server as a configuration server.

Administrators have the flexibility to choose an existing Tivoli Storage Manager server or implement an additional server as a dedicated configuration server. Tivoli Storage Manager servers that obtain information from a configuration server are known as managed Tivoli Storage Manager servers. Policy and configuration information that can be centrally defined includes:

- Client option sets
- Backup, archive and HSM policies
- Client and administration schedules
- Server scripts
- Administration definitions

Policy and configuration information is grouped into one or more named configuration profiles to which managed servers subscribe. Configuration profiles give administrators the flexibility to decide which sets of information are to be administered centrally and which are to be defined locally.

- **Enterprise command routing**

  Enterprise command routing can greatly simplify reporting across a large Tivoli Storage Manager enterprise by allowing administrators to issue one or more commands that perform tasks on a single Tivoli Storage Manager server or group of servers. Results from routed commands are returned to the request origin. A message is also returned indicating whether the commands were successful or whether some or all failed. Command routing in Tivoli Storage Manager provides a simple mechanism for querying and updating multiple servers simultaneously, as you can see from Figure 84.

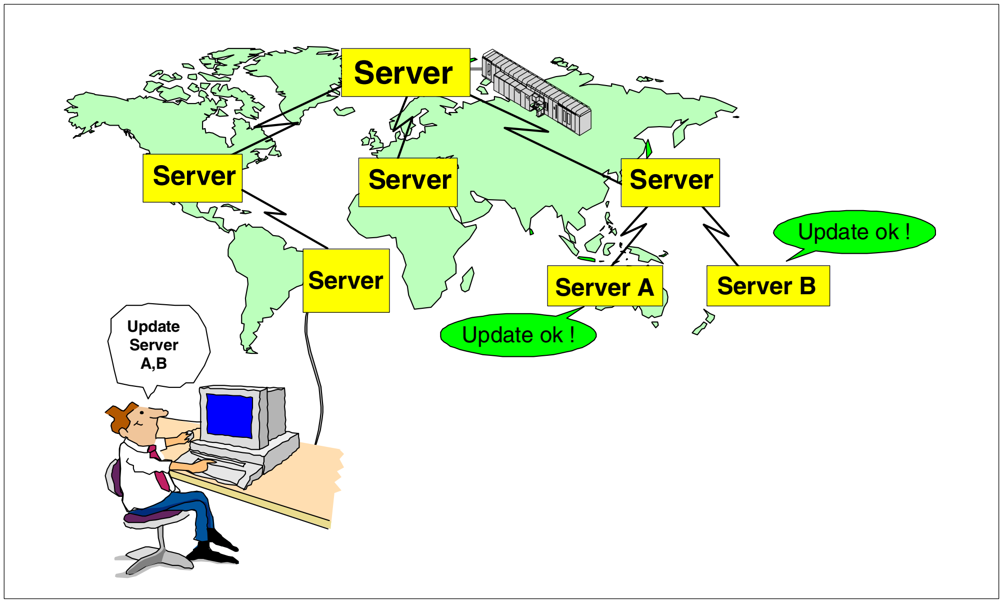


Figure 84. Propagating commands to servers

- **Enterprise logging**

  The enterprise logging function enables more proactive management and better enterprise reporting of a dynamic environment by routing events from groups of Tivoli Storage Manager servers and clients to a single Tivoli Storage Manager server for viewing and reporting. Tivoli Storage Manager servers in an organization can forward their events and those of their clients to a designated Tivoli Storage Manager event server. This event server then displays the events through the Tivoli Storage Manager enterprise console and stores them in the activity log. Events can also be routed to external managers such as SNMP managers, NetView, Tivoli Enterprise Console (TEC) and the Windows NT Event Log.

Chapter 12. Virtual Volumes

Pointer to "how-to" information:

An ITSO "how-to" guide is available as a companion to this conceptual information. It provides material describing how to implement and operate this function. It is available as a hardcopy redbook, *ADSM Server-to-Server Implementation and Operation*, SG24-5244.

The most recent version of the guide is maintained on the ITSO Internet Web pages at:

```
http://www.redbooks.ibm.com/solutions/systemsmanagement/tsm/c_virtualvol.html
```

Tivoli Storage Manager allows a server (a source server) to store the results of various operations on another server (a target server). The data is stored in what are known as *virtual volumes*, which appear to be normal sequential media volumes on the source server, but which are actually stored as archive files on a target server. Virtual volumes can be any of the following:

- Server database backups
- Storage pool backups
- Data that is backed up, archived, or space managed from client nodes
- Client data migrated from storage pools on the source server
- Any data that can be moved by EXPORT and IMPORT commands
- DRM plan files

The source server is a client of the target server, and the data for the source server is managed only by the source server. In other words, the source server controls the expiration and deletion of the files that comprise the virtual volumes on the target server.

At the target server, the virtual volumes from the source server are seen as archive data. The source server is registered as a client node (TYPE=SERVER) at the target server and is assigned to a policy domain. The archive copy group of the default management class of that domain specifies the storage pool for the data from the source server.

> **Note**: If the default management class does not include an archive copy group, data cannot be stored on the target server.

Note that the server to server virtual volume support is only for primary storage pools in the base Tivoli Storage Manager product license. You need to license Tivoli Disaster Recovery Manager to be able to create copy storage pools or server database backups or to generate recovery plan files on another server.

Using virtual volumes can benefit you in the following ways:

- The source server can use the target server as an electronic vault for rapid recovery from a disaster.
- Smaller Tivoli Storage Manager source servers can use the storage pools and tape devices of larger servers.
- For incremental database backups it can decrease wasted space on volumes and limited use of high-end tape drives.

Be aware of the following when you use virtual volumes:

- If you use virtual volumes for database backups, you might have the following situation: SERVER_A backs up its database to SERVER_B, and SERVER_B backs up its database to SERVER_A. If this is the only way databases are backed up and both servers are at the same location, if a disaster strikes that location, then you may have no backups with which to restore your databases.
- A corollary to the first point is that if you are storing objects on a remote server, whether they be database backups, storage pool date or anything else, you must have active communication between the source and target server in order to be able to store and retrieve those objects. The remote server must also be up and running! If you are relying on the source server to perform a disaster recovery, then your first step will be to establish communications between your recovery server and the target server.
- Moving large amounts of data between the servers may slow down your communications significantly, depending on the network bandwidth and availability. High speed, reliable networks are recommended if large volumes of data will be transmitted.
- You can specify in the device class definition (DEVTYPE=SERVER) how often and for how long a time the source server will try to contact the target server. Keep in mind that frequent attempts to contact the target server over an extended period can affect your communications.
- Under certain circumstances, inconsistencies may arise among virtual volume definitions on the source server and the archive files on the target server. You can use the RECONCILE VOLUMES command to reconcile these inconsistencies (see 12.3, “Reconciling virtual volumes and archive files” on page 150 for details).
- Storage space limitations on the target server will affect the amount of data that you can store on that server.
- To minimize mount wait times, the total mount limit for all server definitions that specify the target server should not exceed the mount total limit at the target server. For example, a source server has two device classes, each specifying a mount limit of two. A target server has only two tape drives. In this case, the source server mount requests could exceed the target server's tape drives.

12.1 Setting up source and target servers for virtual volumes

In the source/target relationship, the source server is defined as a client node of the target server. To set up this relationship, a number of steps must be performed at the two servers. In the following example (illustrated in Figure 85 on page 148) the source server is named DELPHI and the target server is named TOKYO.

At DELPHI:

1. Define the target server:
   - TOKYO has a TCP/IP and port address of 9.115.3.583:1845
   - Assign to TOKYO the password CALCITE.
   - Assign DELPHI as the node name by which the source server DELPHI will be known at the target server. If no node name is assigned, the server

name of the source server is used. To see the server name, you can issue the QUERY STATUS command.

2. Define a device class for the data to be sent to the target server. The device type for this device class must be SERVER, and the definition must include the name of the target server.

At TOKYO:

Register the source server as a client node. The target server can use an existing policy domain and storage pool for the data from the source server. However, you can define a separate management policy and storage pool for the source server. Doing so can provide more control over storage pool resources.

1. Use the REGISTER NODE command to define the source server as a node of TYPE=SERVER. The policy domain to which the node is assigned determines where the data from the source server is stored. Data from the source server is stored in the storage pool specified in the archive copy group of the default management class of that domain.
2. You can set up a separate policy and storage pool for the source server.
   a. Define a storage pool named SOURCEPOOL:

   ```
   define stgpool sourcepool autotapeclass maxscratch=20
   ```

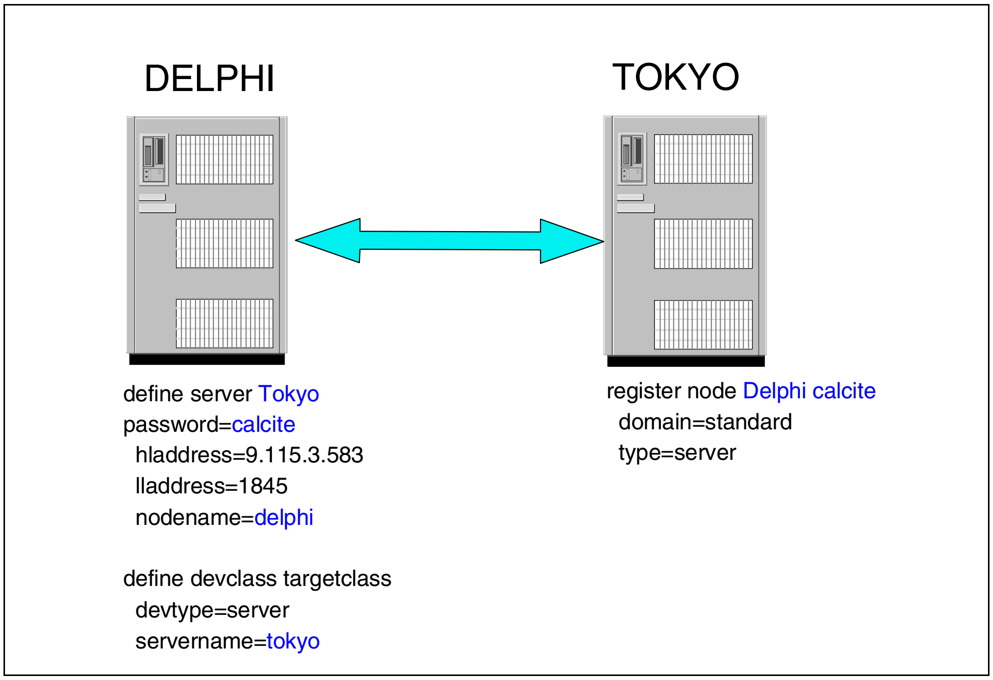
   b. Copy an existing policy domain STANDARD to a new domain named SOURCEDOMAIN:

   ```
   Copy domain standard sourcedomain
   ```

   c. Assign SOURCEPOOL as the archive copy group destination in the default management class of SOURCEDOMAIN:

   ```
   update copygroup sourcedomain standard standard type=archive
   destination=sourcepool
   ```

   After issuing these commands, ensure that you assign the source server to the new policy domain (UPDATE NODE) and activate the policy. See "Changing Policy with the Active Policy Set" in the appropriate *Tivoli Storage Manager Administrator’s Guide* for your operating system platform, for details.



```
define server Tokyo
password=calcite
  hladdress=9.115.3.583
  lladdress=1845
  nodename=delphi

define devclass targetclass
  devtype=server
  servername=tokyo
```

```
register node Delphi calcite
  domain=standard
  type=server
```

Figure 85. Communication configuration for virtual volumes

12.2 Performing operations at the source server

You can perform certain operations at the source server that cause data to be stored in a storage pool at the target server. These operations are:

- Database backups
- Storage pool backups
- Client data backup, archive, or migration
- Data migration from one storage pool to another
- Export of server information
- DRM prepare

The following sections describe how to perform these operations. In the examples, this is assumed:

- The definitions shown in the previous section have been done.
- An operational TCP/IP connection exists between both servers.
- Both servers are running.

12.2.1 Back up the database

You can back up the database of a source server to a target server. For example, to perform an incremental backup of the source server and send the volumes to the target server, issue the following command:

```
backup db type=incremental devclass=targetclass
```

You can also do an automatic database backup to a target server. For example, if you have issued the following command, a database backup occurs automatically when more than 60 percent of recovery log space is used:

```
define dbbackuptrigger devclass=targetclass logfullpct=60
```

12.2.2 Back up a storage pool

You can back up a storage pool from a source server to a target server. For example, a primary storage pool named TAPEPOOL is on the source server. You can define a copy storage pool named TARGETCOPYPOOL, also on the source server. TARGETCOPYPOOL must have an associated device class whose device type is SERVER. When you back up TAPEPOOL to TARGETCOPYPOOL, the backup is sent to the target server. To do this, issue the following commands:

```
define stgpool targetcopypool targetclass pooltype=copy maxscratch=20
backup stgpool tapepool targetcopypool
```

12.2.3 Store client data on a target server

You can configure your Tivoli Storage Manager system so that when client nodes back up, archive, and migrate their data, it is sent to the target server. When clients restore, retrieve, or recall their data, the source server gets the data from the target server. Clients perform these operations just as they normally would.

To configure your system, ensure that the management policy for those nodes specifies a storage pool that has a device class whose device type is SERVER. For example, the following command defines the storage pool named TARGETPOOL.

```
define stgpool targetpool targetclass maxscratch=20
reclaim=100
```

Note: Reclamation of a storage pool automatically begins when the percentage of reclaimable space, which is specified by the RECLAIM parameter, is reached. Reclamation of a target storage pool can involve the movement of a great deal of data from the target server to the source server and back to the target. If this operation occurs automatically during peak operating periods, it could slow network performance significantly. If you set the value to 100, reclamation will not occur automatically. For details about storage pool reclamation and how to begin it manually, see "Reclaiming Space in Sequential Access Storage Pools" in the *Tivoli Storage Manager Administrator's Guide* for your server operating system platform for more details.

12.2.4 Migrate source server storage pool to target server

You can set up your storage pool hierarchy so that client data is migrated from a storage pool on the source server to the target server. For example, storage pool TAPEPOOL is on the source server. The TAPEPOOL definition specifies NEXTSTGPOOL=TARGETPOOL. TARGETPOOL has been defined on the source server as a storage pool of device type SERVER. When data is migrated from TAPEPOOL, it is sent to the target server.

```
define stgpool tapepool tapeclass nextstgpool=targetpool
  maxscratch=20
```

12.2.5 Export server information to a target server

You can use any of the Tivoli Storage Manager EXPORT commands to export data from a source server to sequential media on a target server. You must specify a device class with a device type specified as SERVER. For example, to copy server information directly to a target server, issue the following command:

```
export server devclass=targetclass
```

12.2.6 Import server information from a target server

If data has been exported from a source server to a target server, you can import that data from the target server to a third server. The server that will import the data uses the node ID and password of the source server to open a session with the target server. That session is in read-only mode because the third server does not have the proper verification code.

For example, to import server information from a target server, issue the following command:

```
import server devclass=targetclass
```

12.3 Reconciling virtual volumes and archive files

If you have had to perform a server database restore operation on either the source or target server, you should perform reconciliation between the virtual volumes on the source server and the archive files on the target server. You should also perform reconciliation if you have any other reason to suspect inconsistencies. For example, frequent communication errors between target and source servers could introduce a problem.

To perform reconciliation, issue the RECONCILE VOLUMES command on the source server specifying a device class of the device type of SERVER. You can either specify each SERVER device class in turn or omit this parameter which will reconcile volumes for all SERVER device classes. In the following example TARGETCLASS is a server device class:

```
reconcile volumes targetclass fix=yes
```

The reconciliation action is determined by the value of the FIX parameter as shown in Table 6.

Table 6. Reconciliation actions

FIX=	At the Source Server	At the Target Server	Action
NO	Volumes exist	No files exist.	Report error
		Files exist but are marked for deletion.	
		Active files exist but attributes do not match.	
	Volumes do not exist.	Active files exist.	Report error
		Files exist but are marked for deletion.	None
YES	Volumes exist.	No files exist.	Report error **For storage pool volumes**: Mark volumes as unavailable
		Files exist but marked for deletion	Report error **For storage pool volumes**: If attributes match, mark files on the target server as active again, mark volumes on the source server as unavailable, and recommend that an AUDIT VOLUME be done to further verify the data. If attributes do not match, mark volumes as unavailable.
		Active files exist but attributes do not match.	Report error **For storage pool volumes**: Mark volumes as unavailable and recommend that an AUDIT VOLUME be done to further verify the data.
	Volumes do not exist.	Active files exist.	Mark files for deletion on the target server.
		Files exist but marked for deletion.	None

Chapter 13. Introducing the Tivoli Storage Manager API client

The most recent version of this material is maintained on the ITSO Internet Web pages at:

```
http://www.redbooks.ibm.com/solutions/systemsmanagement/tsm/c_apicli.html
```

The Tivoli Storage Manager application program interface (API) enables an application client to use its storage management functions. It is provided and documented to enable customers or ISVs to interface their own specialized applications to Tivoli Storage Manager. The API can be run in single or multi-threaded mode, which allows applications to create multiple sessions with the Tivoli Storage Manager server within the same process. The API consists of a set of function calls that an application can use to perform the following operations:

- Start or end a Tivoli Storage Manager session.
- Assign management classes to objects before storing them on a Tivoli Storage Manager server.
- Back up or archive objects to a Tivoli Storage Manager server.
- Restore or retrieve objects from a Tivoli Storage Manager server.
- Query the server for information on objects stored there.
- Manage file spaces.

When you, as an application developer, install the Tivoli Storage Manager API, you receive the following:

- The files that an end user of an application would need:
  - API shared library and associated files
  - Sample client options files
  - Documentation
- The source code for the API header files that your application needs.
- The source code for a sample application and the makefile to build it.

The API code is packaged with the backup/archive client code. Applications which have been written using a particular version of the API may not work with later API versions. It is important to check which version API is supported and tested for a particular application. It is possible to use an earlier version API concurrently with a later version backup/archive client if this is required for support of a particular API application.

13.1 Who uses the Tivoli Storage Manager API client?

The Tivoli Storage Manager API client can be used by any application in order to add the Tivoli Storage Manager functions directly into that application. Users can add the API to their program applications to automatically call Tivoli Storage Manager to initiate a backup, restore, archive or retrieve of a file, without having to leave (or close) the application.

This is useful to applications which do not have a Tivoli Storage Manager component available to them, or to applications which need to backup or restore data between processing steps of an application job flow.

13.2 Understanding configuration files and options files

Configuration files and options files permit you to set the conditions and boundaries under which your Tivoli Storage Manager session runs. The Tivoli Storage Manager administrator, the end user, or the developer can set the available options. The values of various options enable the following functions to be performed:

- Initiate the connection to a Tivoli Storage Manager server.
- Control which objects are sent to the server and with what management class they are associated.

On the UNIX platform, the Tivoli Storage Manager options reside in two separate options files, the *client system options file (dsm.sys)* and the *client options file (dsm.opt)*. On other platforms, the options file, *dsm.opt*, contains all of the options. The end user sets up these files when the Tivoli Storage Manager API is first installed on the user's workstation.

The application can also specify options on the *dsmInit()* or *dsmInitEx()* function calls. With Tivoli Storage Manager V3.7, the *dsmInitEx()* incorporates extended version verification for APIs compatibility call. Either use the option string parameter or the API configuration file parameter.

The same option can derive from more than one configuration source. When this happens, the source with the highest priority takes precedence, in the sequence shown in Table 7.

Table 7. Options files

UNIX	Intel
1. *dsm.sys* (client system options)	1. N/A
2. option string (client options)	2. option string (all options)
3. API configuration file (client options)	3. API configuration file (all options)
4. *dsm.opt* (client options)	4. *dsm.opt* (all options)

The different configuration sources, in order of decreasing priority, include:

1. Client system options (UNIX only). Options that a Tivoli Storage Manager or system administrator sets.
2. The API options list takes effect when it is passed to a *dsmInit() or dsmInitEx()* call as a parameter. The list can contain client options, such as:

   compressalways

   servername (UNIX only)

   tcpserveraddr (non-UNIX)

   The API options list permits an application client to make changes to the values of the options in the API configuration file and the client options file. For example, your application might query the end user for the user-preferred format for displaying dates and times. On the basis of the end user's answers, you can construct an API options list with these two options and pass it into the call to *dsmInit()*. You can also set the options parameter to NULL, indicating there is no API options list for this Tivoli Storage Manager session.

3. The values in the API configuration file override the values set in the Tivoli Storage Manager client options file. Set up the options in the API configuration file to have values that you think will be appropriate in the end user's Tivoli Storage Manager session. The values take effect when the API configuration file name is passed as a parameter in the *dsmInit*() call.

   You can also set this parameter to NULL, indicating there is no API configuration file for this Tivoli Storage Manager session.

4. On the UNIX platform, *dsm.opt* contains only the user options file. On other platforms, *dsm.opt* contains all of the options. You can override the options in these files using the methods mentioned above.

13.3 Setting up the API environment

The API uses unique environment variables to locate files. This permits you to use different files for API applications that the interactive client uses. Table 8 lists the API environment variables by platform.

Table 8. API environment variables

Variables	UNIX	Intel	Netware
DSMI_CONFIG	The fully-qualified name for the client option file	The fully-qualified name for the client option file	There are no environment variables. The *dsm.opt, dscameng.txt,* and *dsierror.log* files reside in the same directory as the *dsmapi.nlm* file. This directory becomes the search path for these files.
DSMI_DIR	Points to the path containing dsm.sys, dsmtca, the *en_US* subdirectory, and any other NLS language. The *en_US* subdirectory must contain the *dsmclientV3.cat* file.	Points to the path containing *dscameng.txt*and any NLS message file.	
DSMI_LOG	Points to the path for the *dsierror.log* file.	Points to the path for the *dsierror.log* file.	

Variables	UNIX	Intel	Netware
OBJECT_MODE =64	Support for 64-bit environment (where available) and commmethod is tcpip		

13.4 Using passwordaccess generate without TCA

The Trusted Communication Agent (TCA), a child process which runs on UNIX clients only, normally controls access to the protected password file. It is possible to have the *passwordaccess generate* function without starting the TCA. To do this:

1. Write the application using the *dsmSetUp()* call to pass *argv[0]*, containing the name of the executable. The application is permitted to run as Tivoli Storage Manager-authorized; however, the administrator should decide on the login name for the Tivoli Storage Manager-authorized user.
2. Set the *S* bit (set the effective user ID) to *On* for the application executable. The owner of that executable can then become a Tivoli Storage Manager-authorized user. This permits the user to create a password file, update passwords, and run applications. The owner of the application executable should be the same as the User ID running the program. For example, “User” is User1, the name of the executable is *dapimsp*, and User1 has read-write permissions on the */home/user1* directory.
3. Instruct the users of the application to use the Tivoli Storage Manager-authorized name to log in. Tivoli Storage Manager verifies that the login ID matches the application executable owner before it permits access to the protected password file.
4. Set the *passworddir* option in the *dsm.sys* file to point to a directory where this user has read-write access. For example, under the server stanza in *dsm.sys*, you would enter:

   ```
   passworddir /home/user1
   ```

   The permissions on *dapismp* are:

   ```
   -rwsr-xr-x user1    group1   dapismp
   ```

5. Start the password file and ensure that the Tivoli Storage Manager-authorized user owns the file.
6. Run *dapismp* logged on as User1.
7. Call *dsmSetUp()* and pass in argv.

Note: When running in a multi-threaded mode, and *passwordaccess* is *generate*, only the root, or Tivoli Storage Manager-authorized user, is permitted access, so that the TCA child process is not started.

Part 4. Application and database support

Part 4 covers the ways in which Tivoli Storage Manager interacts with some popular applications.

Chapter 14. AFS and DFS overview

The most recent version of this material is maintained on the ITSO Internet Web pages at:

`http://www.redbooks.ibm.com/solutions/systemsmanagement/tsm/c_afsdfs.html`

Today, businesses work in a global marketplace without time and geographic boundaries. International collaboration, Internet commerce, a mobile workforce, and scarce resources are today's realities. The tie that binds this globally scattered business infrastructure is shared information. To remain competitive in the global marketplace, a company's access to this information must be fast and direct, regardless of where the information is stored or where it is being delivered. As a result, seamless, convenient, secure shared access to information has never been more critical to supporting the business goals of an organization.

IBM and its Transarc subsidiary offer two enterprise file system products—AFS and DFS—that provide a single, shared file space to all authorized users from all possible locations. The file-sharing domain is no longer defined by specific organizations, locations, types of machines or network configurations. With enterprise file systems, heterogeneous server and client machines can be joined together into a global, shared information system. Such an environment provides access to files from multiple heterogeneous platform servers supporting applications, regardless of the location of the file or the end user.

Wherever key information resides and wherever it is needed, AFS and DFS enable the user to access all the information in the enterprise file system, using standard, unmodified, pre-existing applications such as word processors, presentation and publication packages, engineering tools and workflow applications. By seamlessly integrating with these applications, enterprise file systems transform a network of computers into the unified productive workspace required for the global information age.

14.1 Andrew File System

Andrew File System (AFS), the original enterprise file system product from Transarc, has its origins in Carnegie Mellon University's (CMU) Andrew Project. AFS was initially developed to provide central management and storage for file resources across an entire university campus. This environment consisted of hundreds or thousands of heterogeneous workstations used by students and faculty, who needed to share data regardless of which system they were using. In 1989, members of the original design team founded Transarc and commercialized this technology to become AFS.

AFS has since seen widespread success at over 400 worldwide organizations which are using AFS to provide information to over two million end-users and tens of millions of users through the Web. Commonly supported applications include internal publication of reports, policies and procedures, as well as shared workspace for scientific collaboration and software design. AFS is also commonly used to support other large workgroup functions and for software distribution. Furthermore, AFS has also proven to be an excellent infrastructure for highly available World Wide Web content hosting.

AFS is designed for medium-sized businesses and large departments within a corporation that need a scalable, high-performance, reliable distributed file sharing system. AFS provides robust security and an easy-to-use administrative model for file sharing that supports broad platform coverage on UNIX (IBM AIX, Sun Solaris, Sun OS, HP-UX, SGI-IRIX and Digital UNIX), Windows (NT, 95 and 98), and additional desktop platforms such as Macintosh and Netware.

14.2 Distributed File System

Distributed File System (DFS) has its origins in AFS, and was developed as part of the Open Software Foundation (OSF) Distributed Computing Environment (DCE) in the early 1990's. DCE and DFS were the outcome of the OSF's multi-vendor, cooperative initiative to create a set of standard components for distributed computing environments.

DFS has also seen widespread success, within more than 200 worldwide organizations using DFS to provide information to more than one million end-users and to tens of millions of users through the Web. Commonly supported applications are large scale general purpose file sharing within government agencies and global enterprises, Effective Engineering Collaboration within aerospace, electronic and automotive corporations, and Highly Available Web Content Hosting Systems within financial services organizations providing Web-based securities trading and online service offerings.

DFS is designed for tightly integrated enterprise applications where a standard infrastructure, such as DCE or other DCE-enabled applications like IBM's TXSeries Encina API, IBM's Global Sign-On, or IntelliSoft's SnareWorks, are of interest. DFS supports global enterprises and large organizations with enterprise file sharing across the mainframe (OS/390), UNIX (IBM AIX, Sun Solaris and HP-UX), Windows (NT, 95, and 98), and additional desktop platforms, such as Macintosh, Netware, and OS/2. DFS uses the industry-standard DCE security, naming, and remote procedure call (RPC) for its distributed computing infrastructure.

14.3 Tivoli Storage Manager backup support for AFS and DFS

The Tivoli Storage Manager AIX backup-archive client includes support for both AFS and DFS.

The Tivoli Storage Manager AIX AFS/DFS backup support system includes:

- AFS file backup client programs (command line and GUI)
- DFS file backup client programs (command line and GUI)
- AFS volume agent programs
- DFS file set agent programs

The AFS file backup client and the DFS file backup client allow users to back up or restore either AFS or DFS files and directories, as well as the AFS and DFS access control lists (ACLs) for directories, including the additional mount point support. Full support of ACLs means that users backing up data using the backup/archive GUI or the backup/archive command line have all ACLs backed up for possible future restoration. This helps ensure that access privileges are maintained as they were before the failure. Tivoli Storage Manager provides recognition and appropriate handling of mount points. Its ability to exclude unwanted file systems and directories allows Tivoli Storage Manager users to avoid scanning entire DFS directory trees needlessly.

By specifying variable virtual mount points, Tivoli Storage Manager makes it possible to designate many virtual mount points with only one virtual mount point statement. This eliminates the overhead costs that could be accrued by potentially thousands of virtual mount points having to be processed individually.

The AFS volume backup to Tivoli Storage Manager and the DFS file set backup to Tivoli Storage Manager consist of an interface program between either AFS or DFS backup commands and a Tivoli Storage Manager server, to backup and restore AFS or DFS data. The interface program is a fully integrated part of the backup mechanism and uses Tivoli Storage Manager API function calls to perform these tasks. This means that each phase within the backup process can be run in parallel. Tivoli Storage Manager then buffers these multiple "tasks" to disk and migrates the data to tape at high speed, which dramatically decreases the time required to complete backup.

The AFS volume backup agent stores AFS data by volume. The DFS file set backup agent stores DFS data by file set. Through the Tivoli Storage Manager API, each full or incremental volume dump or file set dump is sent to a Tivoli Storage Manager server as a file with the same name as that of the volume or file set. The volume dump files or the file set dump files associated with a dump are stored in a single file space in Tivoli Storage Manager storage called the dump ID string. A Tivoli Storage Manager administrator can delete an AFS or DFS backup dump from Tivoli Storage Manager storage by deleting the file space in which it is stored.

14.3.1 Key features

Tivoli Storage Manager AFS/DFS backup client system support consists of the following programs:

- AFS file backup GUI and command-line clients
- DFS file backup GUI and command-line clients
- AFS volume backup agent
- DFS fileset backup agent

Each program is described in more detail in the following manuals: *Tivoli Storage Manager Using the UNIX Backup-Archive Clients*, SH26-4105, and *Tivoli Storage Manager AFS/DFS Backup Clients,* SH26-4106.

14.3.2 The BUTC and BUTA backup programs

Transarc's AFS backup commands back up AFS volumes to tape devices using the BackUp Tape Coordinator (BUTC) program that prompts for the tapes to which it writes the backup data. Each machine running BUTC is connected to a tape device that can be a simple drive or a stacker.

The Tivoli Storage Manager agent program, BUTA (which was originally derived from *Back Up To ADSM*), is a replacement for BUTC, the tape interface program of the volume backup system. The AFS/DFS BUTA program is an interface between the AFS/DFS backup commands and a Tivoli Storage Manager server, and lets you back up and restore AFS data by volume and DFS data by fileset. AFS/DFS BUTA accomplishes these tasks using Tivoli Storage Manager application programming interface (API) function calls. Through the Tivoli Storage Manager API, each full or incremental volume dump is sent to a Tivoli Storage Manager server as a file with the same name as that of the volume.

14.4 Benefits of using the AFS/DFS BUTA programs

The AFS/DFS BUTA programs offer several benefits over using Transarc's BUTC program to back up or restore volumes/filesets. Table 9 compares the characteristics of the BUTC and the BUTA programs.

Table 9. Comparing the BUTC and the BUTA programs

Characteristics of BUTC	Characteristics of BUTA
Each BUTC instance requires a tape device. It is not always practical to run BUTC on each file server.	The BUTA program provides a single point of control through the Tivoli Storage Manager server for all tapes.
There is no disk buffering.	Disk buffering permits multiple incoming buta streams per tape drive.
It is not practical to use high-performance tape libraries.	The number of BUTA instances you start is limited only by the network constraints, thereby providing the possibility of increased performance.
	You can create backup copies using the Tivoli Storage Manager copy storage pool function.

Chapter 15. Tivoli Storage Manager and databases

The most recent version of this material is maintained on the ITSO Internet Web pages at:

`http://www.redbooks.ibm.com/solutions/systemsmanagement/tsm/c_database.html`

This chapter describes how to use Tivoli Storage Manager to back up databases. An overview of relational databases will also be provided. The fundamental structure of a database, such as tables, table spaces, data files, control files, parameter files, and configuration files will be described. Specific data storage considerations for UNIX-oriented systems, such as using raw devices or file system files, are covered. Techniques for online and offline backup are discussed. Non-relational database management system products, are not specifically discussed, but many of the concepts are also applicable. In subsequent chapters, we discuss techniques for specific database products at a conceptual level. More information on using specific products with Tivoli Storage Manager is contained in the redbook *Using ADSM to Back Up Databases*, SG24-4335, as well as in a number of other specialized redbooks as listed in Appendix B, “Related publications” on page 293.

15.1 Relational databases

The early 1980’s opened the door for relational database management systems (RDBMSs). Since then, these have become so popular that most line-of-business applications being implemented today use RDBMSs. Some people predict that in the beginning of this century, relational databases will have been superseded by object databases. However, whether or not this prediction turns out to be accurate is of no immediate importance to those of us who must back up the data used by today's applications.

RDBMSs share a common set of principles and, conceptually, similar logical and physical structures. Let us look at their fundamental structures, as shown in Figure 86: tables, table spaces, log files, and control files. Please note that although all RDBMS products are based on the same set of principles, they do not all use the same terminology or structures. For example, a table space in Informix is called a dbspace, and there is no table space concept in Sybase or Microsoft SQL Server. Log files in Oracle are called redo logs, while in Sybase they are called transaction logs.

It is important to understand the basic structures of RDBMSs so that you can put an effective backup and recovery strategy in place. You need to back up more than the database itself to ensure a successful recovery.

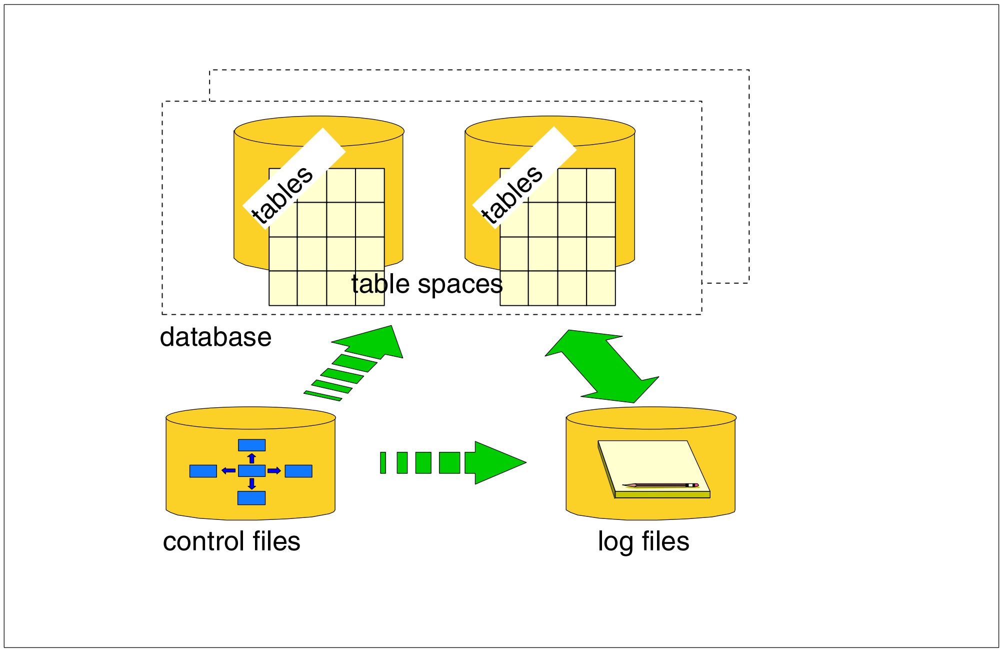


Figure 86. Fundamental structure of a database

15.1.1 Tables

RDBMSs hold their data in the form of two-dimensional tables (sometimes also referred to as relations). These two-dimensional tables are easy for users to understand and manipulate. They also allow different users and applications to view and process the same data in different ways without requiring complex structures.

15.1.2 Table spaces

Table spaces are logical concepts that many RDBMSs use. When a user creates tables in an RDBMS that supports table spaces, the tables are created within a table space. Table spaces provide a convenient way of separating the user's view of data from some of the practical considerations associated with storing that data on disk. In many UNIX environments, table spaces can be implemented using either files or raw devices.

A table space provides the link between the logical view of a database that the user sees and the data files that the database uses to hold the data.

15.1.3 Log files

Most RDBMSs maintain details of updates to databases in log files. If, for some reason, a transaction that makes a change to the database fails to complete successfully, the RDBMS recovery procedure will use the log file to detect that an update may be only partially complete and to undo any changes that the transaction made to the database.

RDBMSs use log files to record the changes made to databases. Log files often can be used to maintain database consistency in the event of an error or failure. Different RDBMS suppliers use different terms for log files.

Some RDBMSs support the use of log files to perform forward recovery, also referred to as roll-forward recovery. Forward recovery takes advantage of the fact that log files hold details of all changes that have been made to the database, and therefore, you do not necessarily need to undo changes, but instead can reapply changes. Log files can be used for forward recovery for both online and offline backup techniques.

Please note that RDBMSs have very complex schemes to manage log files, which we have somewhat oversimplified here. RDBMSs typically have multiple sets of log files to ensure the proper recording of database transactions. Most RDBMSs have a set of online log files as well as offline (or archived) log files. Online log files are used to record the current database transaction activity, and at some point-in-time when the online logs become full, they become offline logs and are moved to another location. Typically backup applications backup the offline log files.

15.1.4 Control files

Each RDBMS needs to hold information about the physical structure of the database, such as which physical files are used by each table space and which is the current log file. We call this information control data. Some RDBMSs (for example, Oracle) hold this data in separate files. Others (for example, Informix and Sybase) hold it within the database itself.

We use the term control files to refer to files that hold control data. For those RDBMSs that hold control data in separate files, you need to define policies for backing up and restoring those files.

15.1.5 Initialization parameter and configuration files

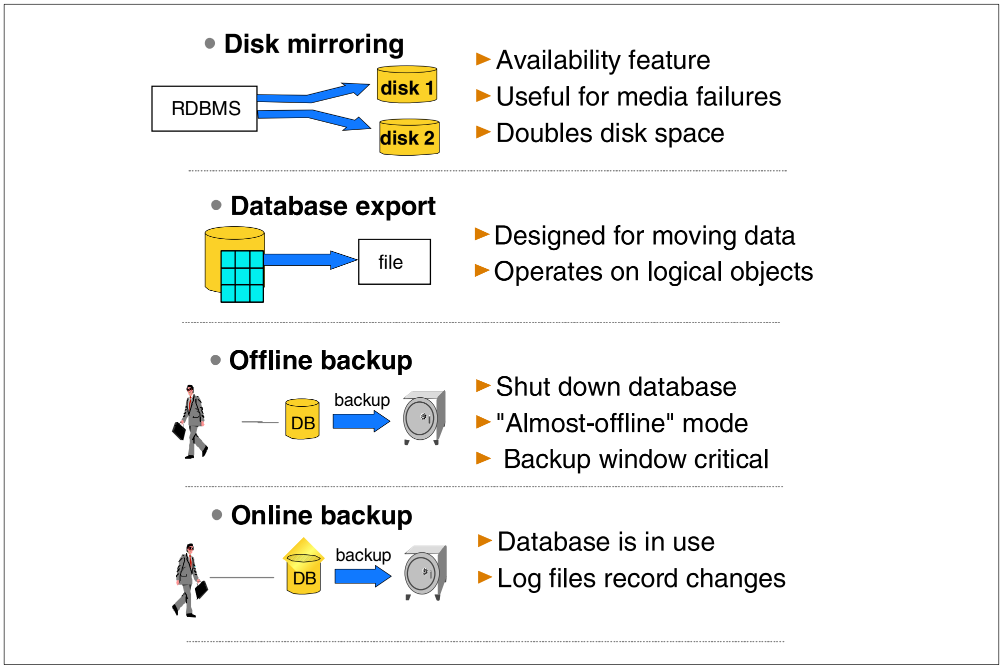
All RDBMSs provide a range of options. Some are set permanently, and others can be modified even when a database is in use (running). Some options allow you to tune the performance of the database; others allow you to specify how you want logging, for example, to be implemented. It is often more convenient to specify the options you require when you start up the database. Most RDBMSs allow you to specify (at database startup time) a file that contains a list of how you want these options set initially. We call these files initialization parameter files. Sometimes we abbreviate and call them initialization files or parameter files.

Installations may have multiple databases, and these databases may have multiple initialization parameter files. One reason for having multiple initialization parameter files for a single database might be to optimize performance for different circumstances. For example, you may decide to allocate one set of values when the database is used for batch processing and another set when it is used for online transactions. Although some of the options are set differently for each situation, many will be the same. Some RDBMSs allow you to specify options that are common to multiple initialization parameter files in configuration files. Instead of repeating all options and their values in each of the initialization parameter files you can select the configuration file that contains the options that you want to use.

You need to define policies for backing up and restoring both initialization parameter files and configuration files.

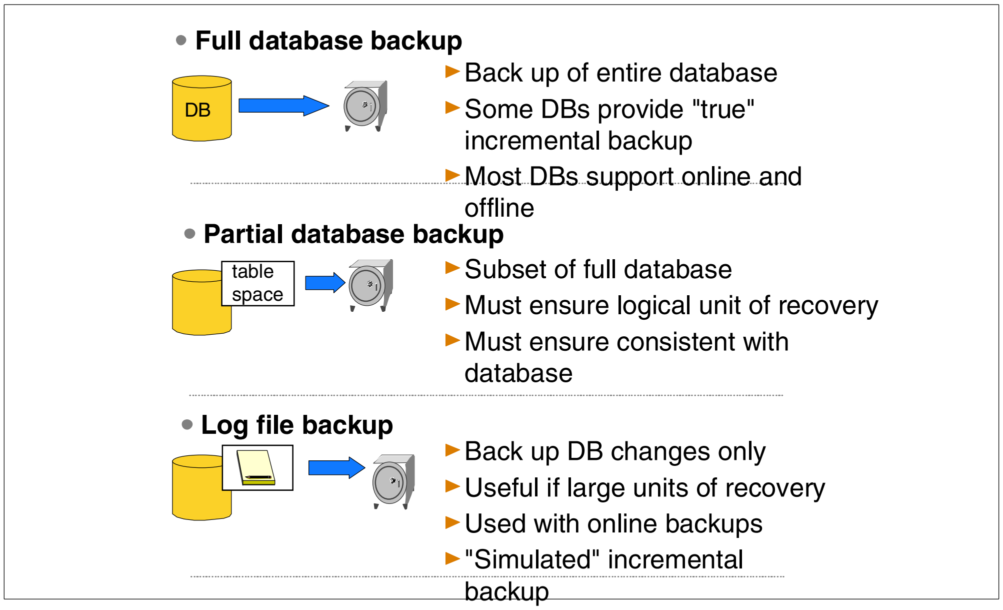
15.2 Backup techniques

There are many techniques you can use to back up data managed by a RDBMS. These techniques are, at least at a conceptual level, common to most RDBMSs. Often, a combination of techniques is used. The techniques we consider are shown in Figure 87 and Figure 88: disk mirroring, database export, offline backup, online backup, full database backup, partial database backup, and log file backup.



- **Disk mirroring**
  - ► Availability feature
  - ► Useful for media failures
  - ► Doubles disk space
- **Database export**
  - ► Designed for moving data
  - ► Operates on logical objects
- **Offline backup**
  - ► Shut down database
  - ► "Almost-offline" mode
  - ► Backup window critical
- **Online backup**
  - ► Database is in use
  - ► Log files record changes

Figure 87. Backup techniques to be considered



- **Full database backup**
  - ► Back up of entire database
  - ► Some DBs provide "true" incremental backup
  - ► Most DBs support online and offline
- **Partial database backup**
  - ► Subset of full database
  - ► Must ensure logical unit of recovery
  - ► Must ensure consistent with database
- **Log file backup**
  - ► Back up DB changes only
  - ► Useful if large units of recovery
  - ► Used with online backups
  - ► "Simulated" incremental backup

Figure 88. Other backup techniques

15.2.1 Disk mirroring

Disk mirroring is a useful technique for maximizing the availability of your database. Mirroring is the process of writing the same data to multiple storage devices at the same time. If a media failure occurs, operations are automatically switched to a mirrored copy. Mirroring allows your users to continue working even though a media failure has occurred.

However, mirroring does not remove the need to back up databases. For example, disk mirroring will not allow you to restore a table that has been lost or damaged as result of user error. Also, although disk mirroring dramatically reduces the impact of media failures, there is still a risk of damage to both sides of the mirror.

Another backup technique related to disk mirroring has recently gained popularity: breaking a disk mirror. This technique breaks the synchronization of one of the disk mirrors and takes a backup copy from that broken mirror. The database is still online and available to users with the remaining mirror or mirrors. We call this technique "simulated online" because the backup is not really an online backup; the backup is taken from a nonfunctional broken mirror, not a running database.

There are several disadvantages to this technique. The backups and restores are being done without the database backup utilities and therefore are always full backups. Ensuring that all necessary data is included in the backups as well as figuring out what is needed for the database restore and recovery are the responsibility of the administrator; it is not an automated procedure, and therefore it is more error prone and requires a higher skill level. In addition, the database is not online to users while the mirror is being broken; a quiesce of the file system and application is required before breaking the mirror. The time to re synchronize the broken mirror with the database after the backup can also be quite extensive; Some customers, for large database systems, have estimated that it would take more than 24 hours to re synchronize the mirrors, which would not be acceptable if you assume a daily backup strategy.

Customers considering the mirror-breaking approach for database backup are typically customers who do not have sufficient spare CPU cycles to run online database backups, and hence are trying to offload some CPU cycles to the storage subsystem. Serious consideration should be given to solving the processor CPU constraint instead of using the mirror-breaking technique, because of the disadvantages we have discussed above.

15.2.2 Database export

Export and import utilities operate on logical objects as opposed to physical objects. For example, you can use an export command to copy an individual table. At some later time, you might want to restore the table, in which case you would use the import command. Export and import are not designed as backup and restore utilities, but instead for moving data, for example, for workload balancing or migration. Export and import are often not integrated with the database’s logging capability, so it is up to you to institute the proper procedures to ensure database consistency. However, because export is usually the only utility that can access individual tables, you may need to use export if you have a requirement for keeping, for example, the last 30 days of each table.

15.2.3 Offline backup

Offline backup involves shutting the database down before you start the backup and restarting the database after the backup is complete. Offline backups are relatively simple to administer, however, they suffer from the obvious but significant disadvantage that neither users nor batch processes can access the database (read or write operations) while the backup is taking place. You need to schedule sufficient time to perform the backup to ensure that the periods when the database will be unavailable are acceptable to your users. Most databases do not require that you perform offline backups if you perform online backups; online backups (along with the log files) are sufficient to recover the database.

Some RDBMSs provide a "single-user mode" or "quiesced mode." You can think of this as an "almost-offline" mode. A database administrator can still use the database, but general users cannot. With some RDBMSs, general users can stay connected to the database but they cannot use it (their transactions are queued). Backup time is reduced with "almost-offline" mode because a full shutdown and restart of the database is not required.

15.2.4 Online backup

Most RDBMSs allow backups to be performed while the database is started and in use (both reads and writes). Clearly, if a database is being backed up while users are updating it, it is likely that the data backed up will be inconsistent. The RDBMSs that support online backup use log files during the recovery process to recover the database to a fully consistent state. This approach requires that you retain the RDBMS log files and indicate to the RDBMS when you are about to start the backup and when you have completed the backup.

Some RDBMSs allow you to quiesce activity on portions of the database (for example, a particular table space) so that a set of complete tables is temporarily "frozen" in a consistent state. You then can back up the set of tables that has been "frozen". Once the backup has completed, you can reactivate the table space.

15.2.5 Full database backup

Full database backups involve making copies of all of the data files used to hold user data. In some database products, full database backups also include copies of the data files that hold tables used by the RDBMS itself, RDBMS log files, and any control files and parameter files that the RDBMS uses. Many RDBMSs allow you to perform full database backups when the database is either online or offline.

Some database products provide incremental backup, which only backs up changed database pages or blocks. This type of incremental backup is called a "true" incremental backup, as opposed to a "simulated" incremental backup (described below as log file backup). The products that support a true incremental backup include Oracle 8, Informix, and a third-party database backup and recovery product, BMC DataTools SQL-BackTrack). Understanding what incremental backup means for a database is a critical concept, so we will explore that in more detail later.

15.2.6 Partial database backup

Many RDBMSs allow partial database backups when the database is online or offline. Partial database backups involve backing up a subset of the full database (such as a tablespace, or data files that make up a tablespace). It is often not the best approach to back up only a subset of a database, because you must ensure that what you back up represents a complete logical unit of recovery from the point of view of the application. Otherwise, you may introduce inconsistencies into the database upon recovery.

15.2.7 Log file backup

For some applications, the units of recovery are too large to be backed up on a daily basis, for example, performing a full daily backup. Sometimes the constraining factor is the elapsed time that is available (the backup window). Sometimes the load that the backup would place on the network would have an unacceptable impact on other processes and users.

In such situations it may be possible to capture only the changes to the database by backing up the RDBMS log files. This type of backup is sometimes referred to as an incremental backup (because you are not performing a full daily backup), but it is really a "simulated" incremental backup. Recovery in such situations is achieved by restoring the database from a full database backup (in some circumstances restoring from a partial backup may be sufficient), restoring the log files, and applying the log files to the restored database (forward recovery). This is a longer recovery process than recovering from a full daily backup or a true incremental backup.

15.2.8 Restore techniques

We have talked in detail about a variety of backup techniques, but we really need to focus on restore techniques as well. As we have discussed, most RDBMS' provide full and partial online backups. Certain types of restores can also often be made while the database is online. Many RDBMSs allow you to restore parts of the database while the rest of the database is online and in use. Typically, these partial online restores are for user data, not system data. You cannot restore an entire database while it is online.

It is also important to distinguish between restoring a database and recovering a database. Restoring a database means bringing back the files that make up the database from your backup system repository. You do not necessarily have an operational database to use at this point. Recovering a database means bringing the restored database to the point of being fully operational, which may, for example, entail restoring log files and performing forward recovery.

15.3 Which backup and recovery technique should you use?

Now that you understand the various backup and recovery techniques, which should you use? Let us compare the backup and recovery technique trade-offs in more detail. Traditionally customers have used a full daily backup technique for their database backups. In between the daily backups, the log files are also backed up to ensure recoverability until the most recent point-in-time. The advantage of full daily backups is simplicity and the possibility of a faster recovery (as compared to true incremental and simulated incremental recoveries). To recover a database (let us say, to the most recent point-of-time) you would restore the last full backup and any logs backed up since the last full backup and then run forward recovery to apply those log files. The significant disadvantage of full daily backups is that they are slower backups. Most RDBMS data is needlessly re backed-up every day even though it has not changed; in typical RDBMS environments the amount of data that changes daily is less than 15 percent!

Simulated incremental backups are faster than full backups because only the log files are backed up daily. Full backups are performed periodically, perhaps once per week, dependent on the activity level in the database and your recovery window requirements. The significant disadvantage of this approach is slower recoveries. Performing forward recovery on the database basically means rerunning all of the transactions in the logs, that is, all of your database activity since the full backup! Forward recovery is a serial process that is totally dependent on the RDBMS. There is nothing that Tivoli Storage Manager can do to speed up this part of the recovery process.

True incremental backup is the best approach if your database has a typical change rate. As a rule of thumb, if your database is changing less than 20 to 30 percent daily (remember that typical databases change less than 15 percent daily), then it is a good candidate for true incremental backups. The first time you run an incremental backup, a full backup is performed. With subsequent backups, only changed database pages or blocks are backed up. RDBMS tools that provide true incremental backup typically offer a variety of incremental backup levels so that you can choose to perform an incremental backup, for example, that captures all changes from the last full backup, or all changes from the previous lower level backup. Keep in mind that true incremental backups for databases use the traditional incremental paradigm, full backups plus incrementals, so periodic full backups (perhaps weekly) are required. The Tivoli Storage Manager paradigm of progressive incremental does not apply to database incremental backups (except for the Tivoli Data Protection for Lotus Notes R4). You would also still back up the log files that are created in between backups, but only the last log file would be required for recovery if you are recovering your database to the most recent point-in-time.

15.4 Exploiting Tivoli Storage Manager

Four techniques can be used to back up databases with Tivoli Storage Manager, as shown in Figure 89. These techniques can involve using the operating system utilities, the RDBMS utilities, and Tivoli Storage Manager combined. Technique (4) is the best technique, because it combines the power of database-aware utilities and the power of Tivoli Storage Manager's superior storage management capabilities. However, if Technique (4) is not available for your database or platform, Techniques (1), (2), and (3) are alternatives.

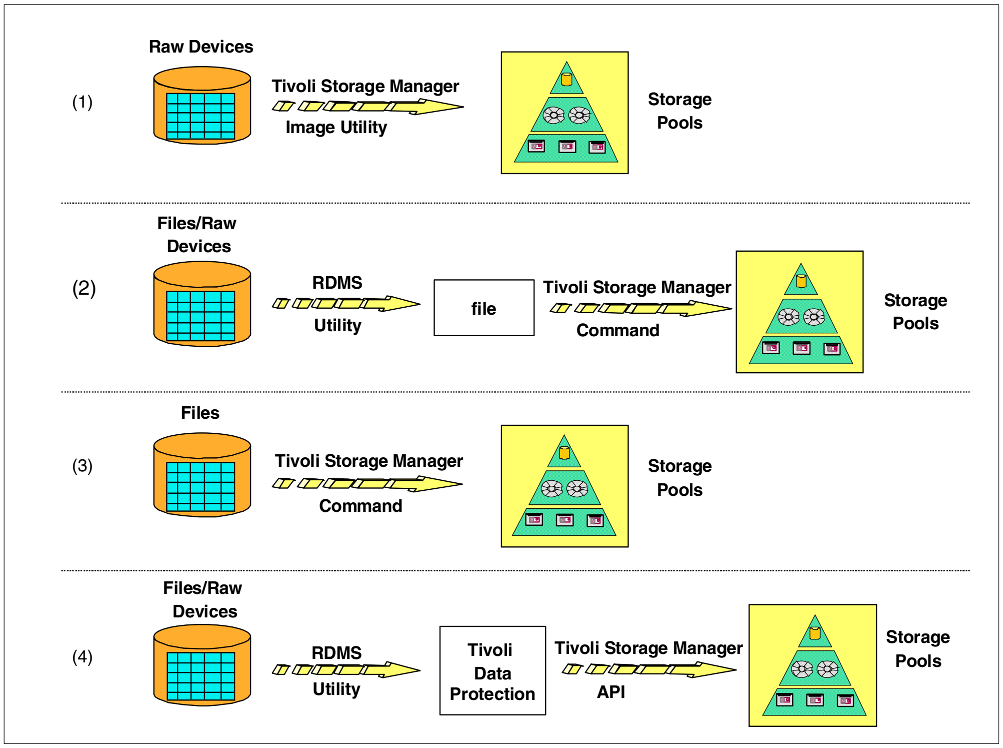


Figure 89. Techniques for using Tivoli Storage Manager to backup databases

Technique (1) applies to databases installed on raw devices. If the database or application has not been enhanced to use the Tivoli Storage Manager's API as in Technique (4), then Tivoli Storage Manager's image utility can be used to back up the raw devices directly. This is similar to using the UNIX dump device (`dd`) command to backup the entire contents of the raw device, but has the advantage of having the Tivoli Storage Manager to manage the backups as well as all the storage space and devices. Technique (1) has the obvious disadvantage that it requires the database to be offline during backup for consistency.

Technique (2) applies to databases installed on files as well as raw devices. You may choose to use an RDBMS utility where available, to back up the data to files and then use Tivoli Storage Manager to back up those created files. By using an RDBMS utility to backup the data to files, there is no need to take the database offline. This allows the database to be available during the backup. This is a better alternative because you will get all the RDBMS utility functionality, which can include online, incremental, and partial backups. This technique requires enough disk space to be available for the intermediate file. If the RDBMS utilities provide backup at a granular level, such as by table space, you might be able to reduce the amount of additional disk space you would need for the files at any one time. You might also be able to send the output to a remotely mounted file system that has more space available than your local database server. Remember also that, when you restore the data that has been backed up by Tivoli Storage Manager, you will need sufficient disk space to hold the files.

Technique (3) applies to databases installed on files. With this technique, you use Tivoli Storage Manager commands to directly back up the files that make up the database and logs. The advantage of this approach is that no intermediate files are created as in (2)—but the database must be offline to ensure a consistent backup copy. Tivoli Storage Manager has no knowledge about the type of files and the data within the files, so we cannot ensure a consistent online backup if you just use Tivoli Storage Manager commands directly. One exception to this is with Oracle. Oracle provides a special utility that allows external tools (like Tivoli Storage Manager) to perform consistent, online backups directly.

Technique (4) applies to databases installed on either raw devices or files. If the application or database product has been enhanced to use the Tivoli Data Protection for Applications (alternatively using SQL-Backtrack utilizing Tivoli Storage Manager API), the underlying physical structure of the database (raw devices or files) is handled by the application, so it does not matter whether raw devices or files are used. Also, the type of backup (for example, online, offline, incremental, table space) is determined and controlled by the application. Technique (4) is the best approach, and therefore this technique is available for all major databases. However, it may not be available for all platforms or database release levels.

Chapter 16. Tivoli Storage Manager and DB2

DB2 is IBM's high-performance, open-platform relational database product. With more than one million server licenses world wide. DB2 is powering the world's e-business solutions by offering open, industrial-strength database management for business intelligence, transaction processing, and a broad range of applications. This chapter focuses mainly on DB2 running in a distributed environment; that is, UNIX, OS/2, and Windows NT. It does not include DB2 for AS/400, nor DB2 for MVS, VM, or VSE. The backup and restore features provided in various versions of DB2 are discussed, with a final section on how these can be integrated with Tivoli Storage Manager.

16.1 DB2 V1

DB2 Version 1 includes backup utilities, both for online and offline, for backing up the whole database. Both a command-line and a GUI is provided.

16.1.1 Online backup

Online backup is done while the database is up and running and in use. In this case, all operations such as create, insert, update, delete, and drop are all permitted.

16.1.2 Offline backup

There can be no other processes connected to the database during offline backup. Each application must be logged off, and when the backup operation starts, no SQL operations are permitted. However, a full shutdown and restart of the database (Stop DBM and Start DBM) is not required.

16.2 DB2 V2

DB2 Version 2 includes significant enhancements to the Version 1 backup/restore utilities. There is a new GUI provided with the Database Director for backup and restore.

16.2.1 Partitioning

With Version 2, databases can be partitioned into table spaces, and backup and recovery can be performed at the table space level. Table spaces can be online or offline during the backup process. Restore must be done with the entire content of the backup image, that is, a table space cannot be restored from a full backup. To restore an individual table space, you must back up individual table spaces. During recovery, all table spaces, other than the one being recovered, can remain online.

16.2.2 Parallel backup and recovery

Another enhancement to the backup/recovery process in DB2 Version 2 is parallel backup and recovery. It is possible to perform the backup or recovery of a database or table space in parallel to or from multiple devices. This is done by establishing multiple paths to or from backup devices such as tape drives or Tivoli Storage Manager. You can specify the actual number of sessions as a parameter of the backup or restore utility. Multiple sessions can drastically reduce the elapsed time requirements.

16.2.3 Multiplexing

DB2 also automatically multiplexes operations, that is, it reads the DB2 data from multiple disks simultaneously and then sends that data across multiple backup sessions. Multiplexing is particularly helpful in improving your throughput if you are backing up straight to tape, and need to more closely match the I/O speeds of the database server disks and the backup server tapes.

16.3 DB2 UDB

DB2 Universal Database (UDB), also known as DB2 Version 5, provides additional significant enhancements over DB2 Version 2. DB2 UDB Version 6 is now available.

16.3.1 DB2 UDB enhanced parallelism

DB2 UDB provides more parallelism in the backup/restore process. Multiple processes or threads can now read data from the database in parallel. Multiple table spaces can be written in parallel during restore.

16.3.2 DB2 UDB enhanced recovery and restart

DB2 UDB also provides additional recovery options, including table space point-in-time recovery and restoration of a subset of table spaces from any backup image. A faster restart, which reduces your total down time during a full restore, is also a capability of DB2 UDB.

16.4 DB2PE

DB2 Parallel Database Architecture was initially implemented in DB2 Parallel Edition (DB2PE) but is now superseded by DB2 UDB Extended Enterprise Edition (EEE). Parallel database technology has been under development at IBM's Almaden Research Center and elsewhere in IBM for several years.

A parallel database must push the envelop on performance and capacity; therefore it must fully leverage the capabilities of the underlying hardware, operating system, and computing environment. DB2PE and DB2 UDB EEE exploit a shared-nothing architecture, as shown in Figure 90 on page 175: Shared-Nothing Parallel Database Architecture, where each node has its own processor, memory, and disk, and effectively "nothing" is shared. A shared-nothing architecture has the best scalability, because most of the processing is done independently on each node, with minimal interaction among nodes.

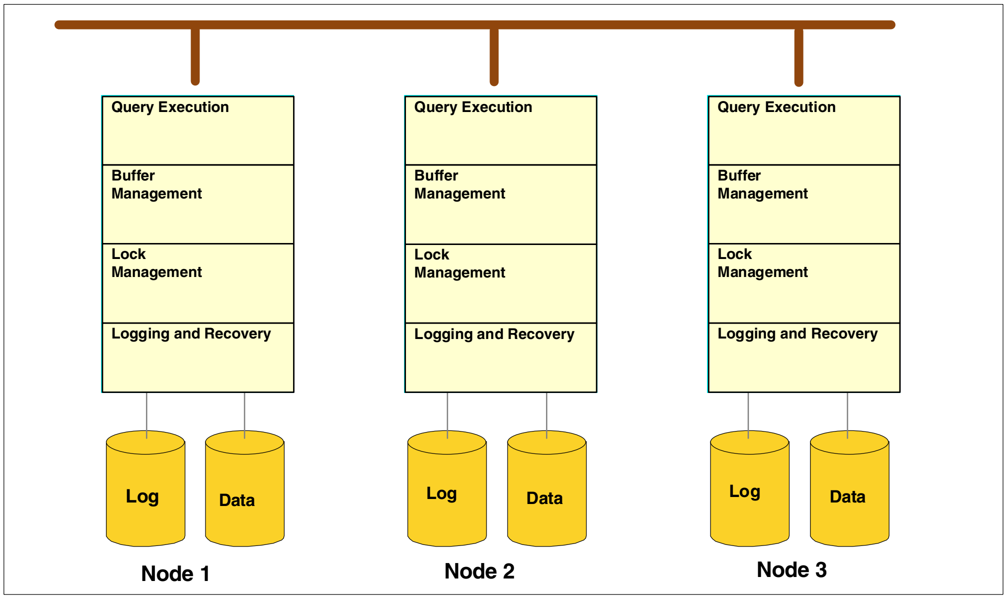


Figure 90. Shared-nothing parallel architecture

Each node in a DB2 Parallel Database system supports a subset of the overall database. For a given user request, data access, buffer management, locking, logging, and other processing is done, where possible, independently on each node in parallel. The database appears to the application user as a single database on a single system.

DB2 Parallel Database exploits the IBM POWER Parallel SP. However, it is in no way tied to or dependent on the hardware platform. DB2 Parallel Database can run as multiple instances on a RISC System/6000 uniprocessor as a means of doing parallel I/O. On an SMP model of a RISC System/6000, multiple DB2 instances can be used to leverage all processors in the single machine. DB2 Parallel Database can also run on separate RISC System/6000s connected together on a LAN. Different RISC System/6000 models can be used, including high-availability HACMP configurations.

DB2PE is based on DB2 Version 1 and therefore has the same level of backup/restore utilities as DB2 Version 1, that is, full online and offline backup. DB2 UDB EEE provides all the rich DB2 UDB backup and recovery functions, including table space backups, advanced parallelism, and a fuller set of recovery options.

DB2 also provides an export/import utility to move data. This utility can be used as a supplement to your backup strategy, but it is not a replacement for the backup utility. There is a risk of introducing inconsistencies into your database because no synchronization of data with logs is performed. However, if you have a requirement to capture individual tables, export may be required because the backup utility only provides either full database or table space-level granularity.

16.5 Integration with Tivoli Storage Manager

As shown in Figure 91, the DB2 backup utilities (but not export/import) are fully integrated with Tivoli Storage Manager services because the DB2 utilities use the Tivoli Storage Manager API. This means that an intermediate file is not created during the backup operation before storing the database image on one or more Tivoli Storage Manager servers. Both online and offline backups can be performed with Tivoli Storage Manager. DB2 data is automatically restored from Tivoli Storage Manager by using the DB2 restore utility.

Tivoli Storage Manager can also archive DB2 log files. DB2 provides a user exit program for backing up and restoring its log files. The user exit program enables DB2 to back up its log files directly to Tivoli Storage Manager. Log files are moved to the DB2 user exit program when they become inactive. The logs are also automatically retrieved from Tivoli Storage Manager for roll forward recovery.

DB2 support of the Tivoli Storage Manager API is provided for databases installed on AIX, HP, SUN, SINIX, OS/2, and Windows NT.

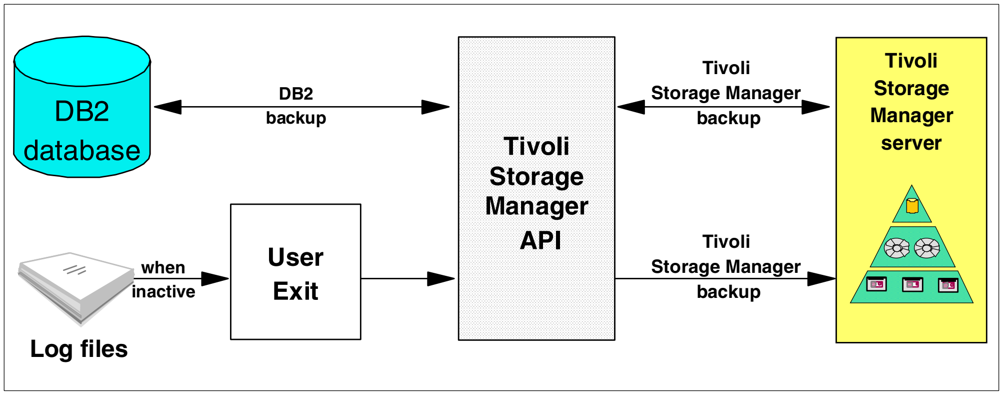


Figure 91. Tivoli Storage Manager interface with DB2

Chapter 17. Tivoli Storage Manager and Oracle

The most recent version of this material is maintained on the ITSO Internet Web pages at:

`http://www.redbooks.ibm.com/solutions/systemsmanagement/tsm/c_oracle.html`

This chapter describes how to back up Oracle databases using both the Tivoli Storage Manager backup client as well as via the application client, which is Tivoli Data Protection for Oracle that can back up Oracle online or offline. The Tivoli Data Protection for Oracle complements the Oracle backup utilities in doing this task by handling all the communications to the Tivoli Storage Manager, letting Tivoli Storage Manager manage the backup to storage devices.

17.1 Tivoli Storage Manager and Oracle

There are three alternatives for online backup of Oracle databases with Tivoli Storage Manager. The first two are discussed in this chapter:

- Oracle 7's Enterprise Backup Utility (EBU) or Oracle 8's Recovery Manager (RMAN) with Tivoli Data Protection for Oracle.
- Customer-written scripts using Oracle `alter table space begin and end backup` commands together with the Tivoli Storage Manager backup-archive client.
- BMC’s SQL-BackTrack product using the Tivoli Storage Manager API. This method is discussed in 25.7, “BMC SQL-BackTrack” on page 235.

Each is a viable alternative, but they differ in functionality and ease-of-use, as well as platform coverage. You could also shutdown the Oracle database and use the Tivoli Storage Manager backup-archive client to backup the files containing the database tablespaces. However, this is not an online backup and is not discussed further here.

17.2 Backup using Tivoli Data Protection for Oracle

Tivoli Data Protection for Oracle supports both Oracle 7 databases with the Enterprise Backup Utility (EBU) and Oracle 8 databases with the Recovery Manager (RMAN). EBU for Oracle Version 7 and RMAN for Oracle Version 8 are utilities provided by Oracle for performing backup (online and offline) and restore (online and offline) of Oracle databases. They can be used only by interfacing them with an external media manager (such as, Tivoli Storage Manager) via the Oracle provided library. After the Oracle7 EBU or Oracle8 RMAN initiates a backup or restore, Tivoli Data Protection for Oracle acts as the interface to Tivoli Storage Manager. It translates Oracle API calls to Tivoli Storage Manager API calls to the Tivoli Storage Manager server for the desired functions. These are typically carried out while users are working — causing minimal disruption.

Tivoli Data Protection for Oracle (also called the Oracle Backup Agent) works in conjunction with the Tivoli Storage Manager API client and should be installed wherever the Oracle databases are located. The output of TDP for Oracle can be sent to any Tivoli Storage Manager server. TDP for Oracle is available for AIX, NT (Intel platform), Windows 2000, Solaris and HP-UX. Note that EBU is only released with Oracle7 and RMAN is released with Oracle8.

See Figure 92 for an understanding of how Oracle7 or Oracle8 works in conjunction with Tivoli Data Protection and the Tivoli Storage Manager server.

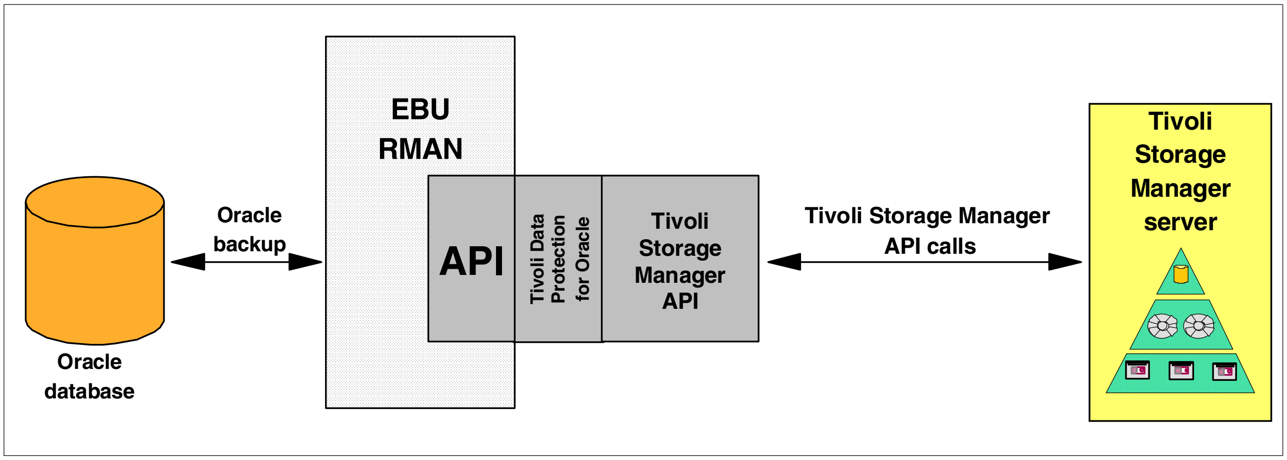

Oracle database
Oracle backup
EBU
RMAN
API
Tivoli Data Protection for Oracle
Tivoli Storage Manager API
Tivoli Storage Manager API calls
Tivoli Storage Manager server

Figure 92. Oracle7 and Oracle8 interfacing with Tivoli Storage Manager

Oracle7 and Oracle8 provide backup and restore functions for Oracle databases. You can perform full or partial, offline (termed as cold backup in Oracle7) or online (termed as hot backup in Oracle7) backups. Once you identify which database you want to back up, Oracle locates all of the necessary files and sends them to Tivoli Storage Manager. Recovery is similarly managed, in that Oracle issues the appropriate commands to Tivoli Storage Manager to restore the required objects (such as tables, control files, recovery logs) and then perform the recovery.

Once the interface to Tivoli Storage Manager is configured, all administrative commands, such as backup, restore and so on, are entirely within the Oracle backup utilities via the EBU or RMAN scripting controls without the need for Oracle DBA knowing Tivoli Storage Manager commands. The Oracle DBA takes control when the backups are to be removed from Tivoli Storage Manager using commands or utilities (for example, ebutool with Oracle7) within Oracle. When using TDP for Oracle, it is important that the DBA is familiar with the underlying utility (that is, RMAN or EBU) and is able to create and maintain the correct scripts for backup and recovery. The advantage of TDP for Oracle is that the full power of these Oracle-provided utilities can be used and it exploits the detailed in-depth knowledge that Oracle has about its database structure. You can backup the Oracle logical entities themselves (for example, individual tables) rather than the file structure and incremental database backup that is also provided with RMAN.

17.3 Oracle alter table space utility

Oracle for many years relied on external tools for backup before providing utilities such as EBU (Oracle7) and RMAN (Oracle8). However, Oracle also provided a way for external tools to perform online backups. Oracle's `alter table space begin backup` command puts a table space into backup mode in such a way that an external tool can back up the table space while it is changing. Changes to the table space are also recorded in the redo log files. You can think of this command as a way for Oracle to logically "freeze" the table space such that the database has knowledge of where the backup begins. Roll-forward recovery provides the consistency for any updates that occur after the table space is frozen. You can write scripts, as shown in Figure 93, that contain an `alter table space begin backup` command, Tivoli Storage Manager client backup commands, and `alter table space end backup` command, for each table space. The client backup commands would backup either the files which contain the database tablespaces directly if the Oracle database is installed on file systems, or else the logical volumes, using the image backup commands if the Oracle database is installed on raw devices. As well as backing up the tablespaces themselves, you also backup the archived redo log files. This operation is typically performed at regular intervals throughout the day.

Although using Oracle's alter table space commands is a viable online backup alternative, it has the disadvantage that you are limited to full, table space, data file, and log file backups. Only simulated, not true, incremental backups are possible. Much of the responsibility of managing the backup and recovery process is still left to each customer to implement. An advantage of this alternative is that it is supported on any platform on which Tivoli Storage Manager provides a backup/archive client, whereas, TDP for Oracle is limited to selected platforms.

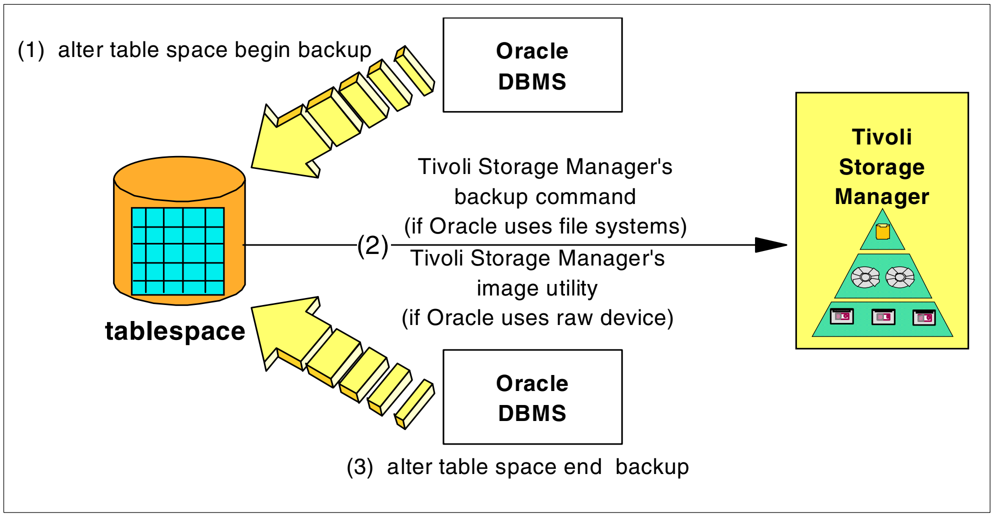

(1) alter table space begin backup
Oracle DBMS
Tivoli Storage Manager's backup command (if Oracle uses file systems)
(2)
Tivoli Storage Manager's image utility (if Oracle uses raw device)
tablespace
Tivoli Storage Manager
Oracle DBMS
(3) alter table space end backup

Figure 93. Alter table space script backup option

Chapter 18. Tivoli Storage Manager and Informix

The most recent version of this material is maintained on the ITSO Internet Web pages at:

```
http://www.redbooks.ibm.com/solutions/systemsmanagement/tsm/c_informix.html
```

This chapter describes the details for backing up INFORMIX-OnLine databases. The various versions of Informix have always contained excellent backup utilities. With Version 7 and later releases, these utilities can be easily interfaced to the Tivoli Storage Manager client API using the TDP for Informix application client. This product contains a library which implements the Open Group Backup Services API (XBSA) functions provided by Informix. The TDP for Informix library allows you to backup and restore Informix server database and logical logs to and from the Tivoli Storage Manager. Informix uses its On-Bar utility to manage its backup on the Tivoli Storage Manager.

TDP for Informix is available for AIX and Solaris platforms today. The following Informix servers are supported:

- Informix 7 - Informix Dynamic Server (IDS)
- Informix 8 - Extended Parallel option (AIX only)
- Informix 9 - Universal Data Option (UDO)

We will look at the provided INFORMIX-OnLine backup utilities, and then go to the specific examples of how to integrate TDP for Informix into your backup strategy, including level-0, level-1, level-2, logical log, and specific dbspaces backups. We also show how to save copies of tables by using INFORMIX-OnLine export utilities. We include examples of how to use the latest INFORMIX-OnLine backup utility, ON-Bar, to back up databases to Tivoli Storage Manager through TDP for Informix. Please note that when we refer to INFORMIX-OnLine in this book we are referring to the INFORMIX-OnLine database server.

BMC also provides a version of their SQL-Backtrack program for Informix. See 25.7, “BMC SQL-BackTrack” on page 235 for more information.

18.1 INFORMIX-OnLine provided backup utilities

Several INFORMIX-OnLine utilities, which have been made available at various release levels, can be used for backup purposes. We will give some examples of backup utilities using INFORMIX-OnLine Version 5.0, INFORMIX-OnLine Version 6.0, INFORMIX-OnLine Version 7.1, and INFORMIX-OnLine Version 7.2. Before we discuss specific backup utilities, it is important to understand a critical difference in terminology between the INFORMIX-OnLine utilities and the storage manager (Tivoli Storage Manager). When data files are sent from INFORMIX-OnLine to Tivoli Storage Manager, regardless of the utility used to initiate backup, all data is, by default, stored in backup copy groups. From the perspective of a Tivoli Storage Manager administrator, this data is considered “backup” data, not “archive” data.

An **archive** (in INFORMIX-OnLine terminology) backs up all of the data that the database server manages, which includes all dbspaces. The archive is written on a tape or to a file and includes all used disk pages. It does not include configuration files, so you have to back them up separately. Archiving is rather sophisticated in that it provides for true incremental archives so that you do not have to archive the entire database each time. Three levels of archive are provided:

- Level-0, the baseline archive, because all used pages are archived
- Level-1, all changes since the last level-0 archive
- Level-2, all changes since the last level-1 archive

You build a schedule for incremental archives that fits the needs of your environment. For example, you can perform level-0 archives monthly, level-1 archives weekly, and level-2 archives daily, or you can build a more frequent schedule. Level-1 and level-2 backups are cumulative since the previous level backup. Given how INFORMIX-OnLine handles incremental backups, we recommend that you carefully consider the Tivoli Storage Manager copygroup options, particularly those that determine the number of data versions that exist, retention periods for extra data versions, and copy mode. The main INFORMIX-OnLine Version 5.0 backup utility is *tbtape*. In INFORMIX-OnLine Version 6.0 tbtape is replaced by *ontape*, which contains enhancements over tbtape. INFORMIX-OnLine Version 6.0 also provides a new backup utility called *ON-Archive* which provides further enhancements over ontape.

We also look at the new INFORMIX-OnLine Version 7.21, 8.0, and 9.0 utility, *ON-Bar*. ON-Bar replaces the functions of ON-Archive and ontape and provides an interface to a storage manager application, such as Tivoli Data Protection for Informix. You can use other INFORMIX-OnLine utilities for backup purposes, but they are actually data migration utilities and not a substitute for tbtape, ontape, ON-Archive, or ON-Bar. The *onunload* (or *tbunload* in INFORMIX-OnLine Version 5.0) and *dbexport* utilities are not coordinated with the information stored in logical log files, and they do not save a copy of system overhead information, which is important to INFORMIX-OnLine.

18.2 The tbtape utility

The main backup utility for INFORMIX-OnLine Version 5.0 is *tbtape*. It is used to back up logical logs (including continuous backups), change the database logging status, create and archive, or restore data.

Continuous logical log backups automatically back up each logical log as it becomes full. Database logging is the process of writing the transactions to the log files. Different types of logging are supported, such as buffered and unbuffered. The tbtape utility is used to turn database logging on and off, and to change the type of logging used. A tbtape archive (in INFORMIX-OnLine terminology) backs up all of the data managed by the database server, which includes all dbspaces. This is called a whole system archive. This archive is written on a tape or to a file and includes all used disk pages. The archive does not include configuration files, so you need to back them up separately. The tbtape archive is performed when the database is in online or quiescent mode. Quiescent mode can be thought of as an “almost-offline” or single-user administrative mode. The tbtape restore is performed when the database is in offline mode.

18.3 The ontape utility

One of the main backup utilities in INFORMIX-OnLine Version 6.0 is *ontape*. It is similar to tbtape, but provides some significant enhancements. With ontape you can restore specific dbspaces, and the restore can be either warm or cold. Warm restore means that INFORMIX-OnLine can be online, and cold restore means that INFORMIX-OnLine must be offline. Warm restores can be done for noncritical dbspaces, that is, any dbspace other than the root dbspace and the dbspaces that contain the physical and logical logs. Please note that the backup operation is still done in online or quiescent modes and still includes all dbspaces. Only the restore has changed to allow specific dbspaces and online mode. Another difference between ontape and tbtape is that with ontape you no longer can use the INFORMIX-OnLine onmonitor utility to create the archive. You can only execute the ontape utility from the command line.

18.4 ON-Archive

ON-Archive is a powerful new utility for INFORMIX-OnLine Version 6.0. It provides the function of ontape, but adds significant enhancements including:

- Archive of dbspace sets (named collectives of one or more dbspaces)
- Archive and restore of different dbspaces in parallel

It provides online and quiescent mode backup and online and offline restore for one or more dbspaces in parallel. You group one or more dbspaces into dbspace sets, and ON-Archive works against these dbspace sets. The output produced by ontape and ON-Archive is incompatible. You cannot back up data with ontape and restore it with ON-Archive (or vice versa).

18.5 ON-Bar

ON-Bar (Backup and Restore) is new to the INFORMIX-OnLine product line (in Versions 7.21 and later) and enhances the backup and restore facilities. ON-Bar replaces ON-Archive and ontape and provides the following enhancements:

- It simplifies backup and restore procedures compared to ON-Archive. Please note that ON-Bar refers to a backup as a backup, not an archive, whereas ON-Archive refers to a backup as an archive.
- It allows for warm backup and restore of dbspaces.
- It enables automatic logical log backup through an ALARMPROGRAM variable that calls a custom-made script.
- It provides automation and management facilities to reduce operator intervention.
- It supports SMP, XMP, and distributed data environments.

ON-Bar is designed to support the X/Open Backup Services API (XBSA). Therefore, any storage manager product that supports this standard works with ON-Bar. The term “storage manager” is used here to denote any type of storage management tool, for example, Tivoli Storage Manager. Therefore, ON-Bar backups can be sent directly to Tivoli Storage Manager through the API; no intermediate file is created.

18.5.1 Components

The Tivoli Storage Manager interface to ON-Bar consists of the following components as illustrated in Figure 94: the ON-Bar program, the X/OPEN Backup Services Application Programmer's Interface (XBSA), the storage manager (using TDP for Informix), and the ON-Bar catalog tables, message file, and emergency boot file.

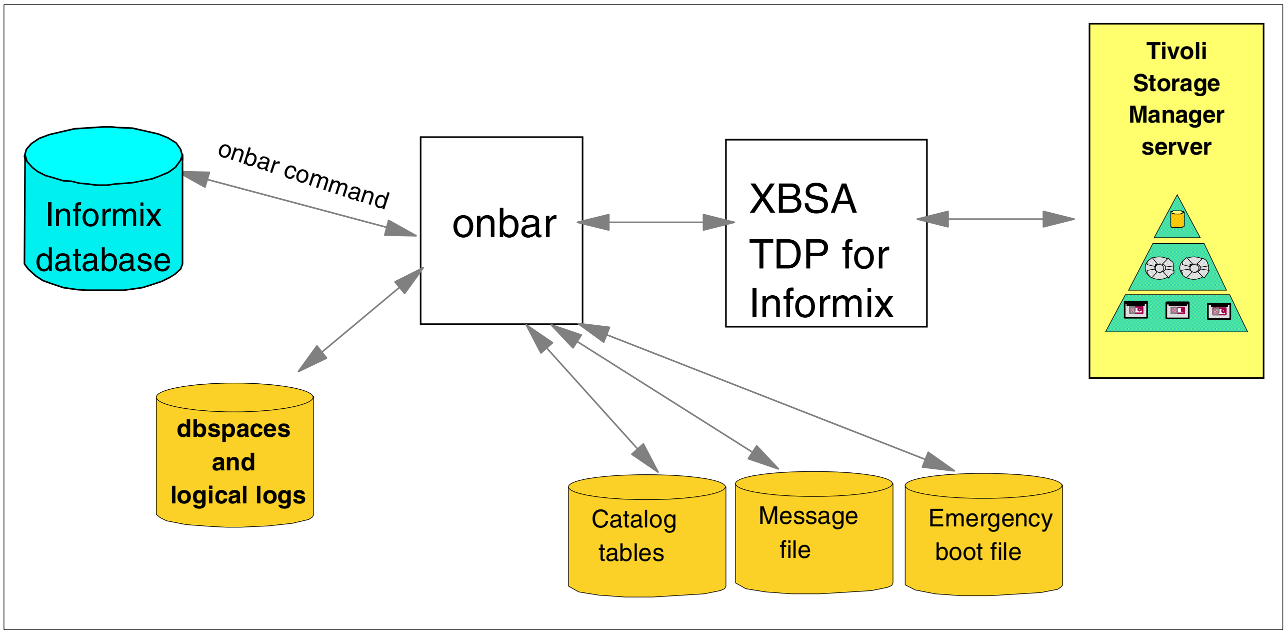


Figure 94. TDP for Informix integration with Informix

18.5.1.1 ON-Bar program

The ON-Bar program receives requests to perform backups and restores and passes them to INFORMIX-OnLine or the storage manager. Users can initiate requests either manually, by issuing a command line request to the ON-Bar program from the client machine where the INFORMIX-OnLine database is stored, or automatically, by using the Tivoli Storage Manager's scheduler to invoke ON-Bar backup or restore. The ON-Bar utility itself does not provide an independent GUI.

18.5.1.2 Storage manager

Tivoli Storage Manager, which is used here as the storage manager, handles the actual data storage and media. It may also provide additional capabilities, such as the ability to:

- Schedule administrative tasks
- Support a distributed environment
- Enable data compression and decompression

18.5.1.3 XBSA

ON-Bar and Tivoli Storage Manager communicate with each other through TDP for Informix. ON-Bar uses the XBSA to exchange backup, restore and control data via TDP for informix, with Tivoli Storage Manager.

18.5.1.4 Catalog tables

The ON-Bar catalog tables, stored in the *sysutils* database, track compatibility of component versions and contain details about dbobjects. There are five tables that contain version and backup information:

- The *bar_server* table tracks instances of the OnLine Dynamic Server.
- The *bar_object* table tracks backup objects.
- The *bar_action* table tracks all backup and restore attempts against each backup object.
- The *bar_instance* table describes each object that is backed up during a successful backup attempt.
- The *bar_version* table lists compatible versions of ON-Bar, storage managers, and XBSA.

The INFORMIX-OnLine Dynamic Server Backup and Restore Guide documents the field contents for each of the above tables in more detail.

18.5.1.5 Message file

The ON-Bar message file contains information about ON-Bar progress as it executes. The file also provides time stamps on various operations that are useful when you are trying to determine how much time the backups will require to complete so that you can develop Tivoli Storage Manager schedules.

18.5.1.6 Emergency boot file

The ON-Bar emergency boot file contains enough information to cold restore an online server instance. The boot file is no longer just for critical dbspaces. It is used, instead of the *sysutils* tables, during a cold restore of the database.

Chapter 19. Tivoli Storage Manager and Sybase

The most recent version of this material is maintained on the ITSO Internet Web pages at:

`http://www.redbooks.ibm.com/solutions/systemsmanagement/tsm/c_sybase.html`

This chapter describes how to use the Tivoli Storage Manager backup-archive client to back up Sybase databases. BMC's SQL-Backtrack is also available for Sybase, as described in 25.7, "BMC SQL-BackTrack" on page 235.

19.1 Sybase native backup/recovery utilities

Sybase provides basic native backup and recovery utilities. The Sybase backup utilities are `dump database` and `dump transaction`, and the corresponding restore utilities are `load database` and `load transaction`. Dump database takes database images of the entire database, including both the data and transaction logs. It allows dynamic dumps; users can continue to use their applications to query or update database data while the dump takes place, that is, it is an online dump utility.

The dump transaction copies the transaction log, providing a record of any database changes made since the last database or transaction log dump.

Sybase also provides an export/import utility, called the bulk copy utility, which is not designed as a backup utility but can be used to supplement your backup strategy. The bulk copy utility is used to transfer data between the server and operating system. It is frequently used to copy data into another program, such as another DBMS or spreadsheet.

The bulk copy utility is the only Sybase mechanism to copy data at a table level; the dump utilities work at the database level. Neither the bulk copy utility nor the Sybase dump utilities are enabled to use the Tivoli Storage Manager API.

19.2 Using Tivoli Storage Manager to back up Sybase

There are two ways in which Tivoli Storage Manager can be used to back up Sybase databases. Since these use the Tivoli Storage Manager backup-archive client, they are both available for any supported client platform.

- You can back up Sybase online, using the `dump database` command and the Tivoli Storage Manager backup-archive client. To perform an online backup, you need to use the Sybase backup utility to create the intermediate backup file. Then you use the Tivoli Storage Manager backup/archive client to back up the file to Tivoli Storage Manager. This method is useful where the database is installed on either filesystems or raw logical volumes, since the Sybase utility extracts the database data into an operating system file which is then backed up. For restore, the reverse process is used. First use the Tivoli Storage Manager client to restore the file, then the Sybase `load database command` to re-create the database itself.

- Tivoli Storage Manager client can be used to back up a Sybase database directly if the database is installed on a file system, and with the image utility (where available) if the database is installed on a raw device. However, note that this backup can only be taken when the database is offline.

Chapter 20. Tivoli Storage Manager and SAP R/3

The most recent version of this material is maintained on the ITSO Internet Web pages at:

`http://www.redbooks.ibm.com/solutions/systemsmanagement/tsm/c_sap.html`

SAP (Systeme, Applikationen und Produkte in der Informationsverarbeitung) founded in 1972 by five former IBM employees, is one of the world's largest software companies providing packaged commercial applications. It now has 28 subsidiaries and affiliates in all of the major industrialized countries of Europe, North America, Asia, and Africa, and it supports more than 3500 customers in 36 countries worldwide. IBM and SAP formed an alliance in July 1983 by signing an international agreement that covers mutual cooperation in development, marketing, sales, and support of customer business solutions. SAP R/3 is an integrated client/server package covering accounting, human resources, logistics, and production planning. R/3 also provides an application development environment. As shown in Figure 95, a typical SAP R/3 system consists of three tiers: database servers, applications servers, and presentation clients.

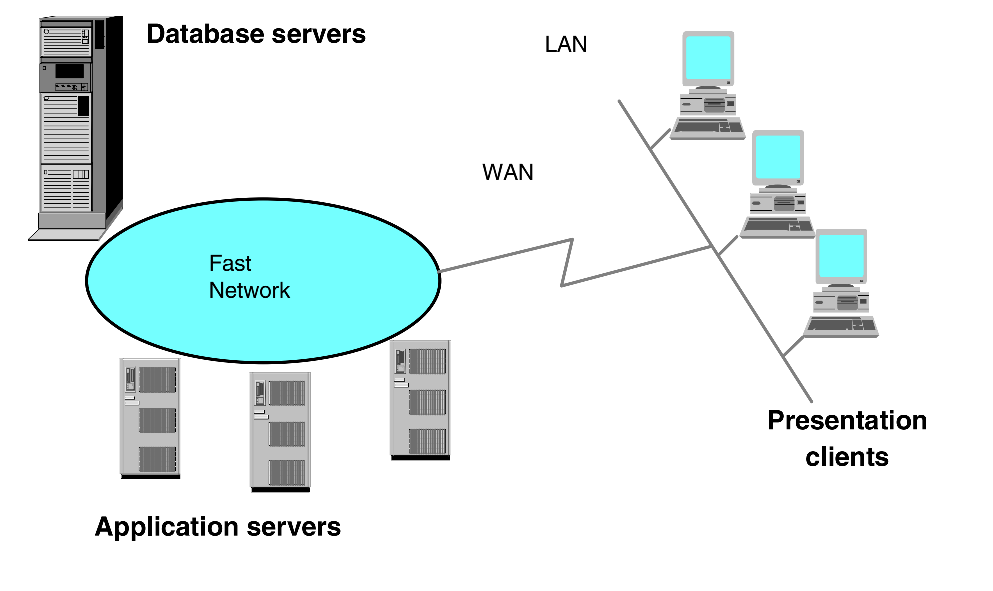


Figure 95. SAP R/3 components

20.1 Fast, reliable, online backup for all major databases

In a SAP R/3 configuration a single application server can reside on the same machine as the database server, or a dedicated database server and one or more application servers can run on separate machines. In the second case, the servers are connected through a fast LAN or a high-performance switch. The end user is connected to SAP R/3 through a presentation front-end with a GUI, the presentation client, typically a PC or X-terminal. The presentation client uses SAP R/3 transactions to send requests to the application server. The application server reads data from local memory (buffered data) or initiates database requests to the database server, which represents the highest level in the server hierarchy. Typically an application server is dedicated to a group of users, such as a department. The SAP R/3 software, certain application development programs, and user data are stored on the database server. SAP has chosen a centralized database concept because it is easier to maintain. SAP R/3 runs on many platforms, including AIX, Compaq TRU64, HP-UX, SNI, Solaris, OS/390, Windows NT, and AS/400 systems. The following database products are supported in an SAP R/3 environment, although, not all databases are available for all platforms. Consult SAP marketing for more information on specific availability.

- DB2
- Oracle
- Informix
- ADABAS-D
- Microsoft SQL Server
- DB2/400

20.2 Methods of backing up R/3 databases

There are several methods of backing up databases in a SAP R/3 environment, but SAP typically supports and certifies only specific methods, which are a subset of all possible methods:

- For DB2, the database utilities (which use the Tivoli Storage Manager API) are supported.
- For Informix, the ONTape, ONArchive, and ONBar utilities are supported. The ONBar utility uses an X/OPEN API. Tivoli Data Protection for Informix provides an X/OPEN API to the Tivoli Storage Manager server via Informix’s ONBar utility.
- For ADABAS-D, the ADABAS-D backup utilities are supported. An external interface to storage management products is provided through ADINT, which interfaces directly with the Tivoli Storage Manager API. It is available as a service offering. For more information, see:

  `http://www.de.ibm.com/entwicklung/adint_adsm/index.html`
- For Microsoft SQL Server, the Microsoft backup/restore functions are supported. Tivoli Storage Manager provides an interface to these functions with the Tivoli Data Protection for Microsoft SQL Server.
- For Oracle, SAP provides their own Oracle backup utilities, BRBACKUP and BRARCHIVE through the SAP DBA interface. They also provide an interface, BACKINT to external storage manager products. The application client, Tivoli Data Protection for R/3 uses the BACKINT interface to connect the R/3 backup utilities to Tivoli Storage Manager. Figure 96 shows the flow that BACKINT uses for backing up Oracle databases when integrated with Tivoli Storage Manager. This product is described in greater detail in “Tivoli Data Protection for R/3” on page 191.
- IBM offers another SAP-related product, ARCHINT/TSM, that interfaces to SAP's archiving functions (for any SAP R/3 database). ARCHINT/TSM is available as a service offering. ARCHINT/TSM allows data from SAP's archive function to be stored and managed by one or more Tivoli Storage Manager servers.

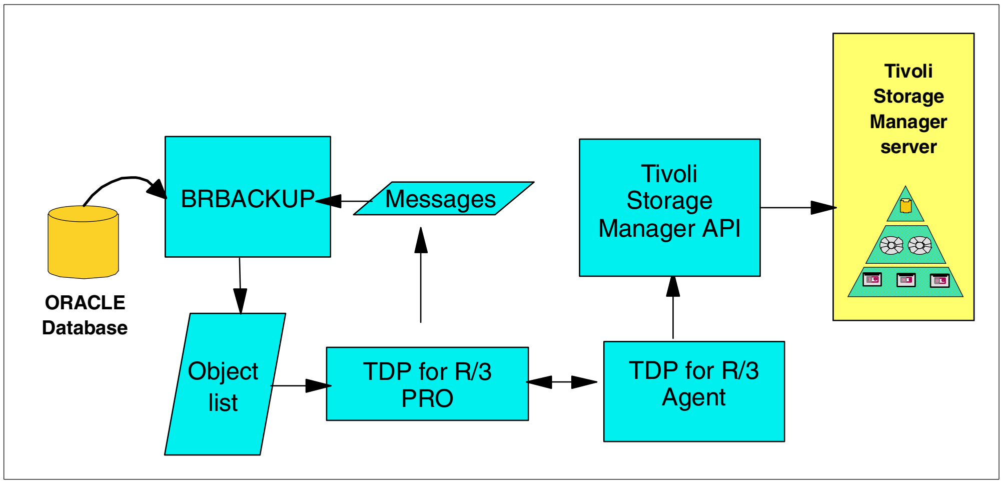


Figure 96. Logical flow of BACKINT operation

Table 10 summarizes the backup alternatives for all supported databases in a SAP R/3 environment.

Table 10. SAP R/3 backup alternatives with Tivoli Storage Manager

Database	Database Implementation	Database Mode	Intermediate File Created	Supported Backup Alternative	Platforms Supported
Oracle OPS	File system Raw device	Offline, Online	No	BACKINT using TDP for R/3/	AIX,HP-UX, Tru64, RedHat Linux,NT, Solaris, Windows 2000
Informix - Online	Raw device	Offline, Online	No	Ontape, Onarchive Onbar using TDP for Informix	AIX, Solaris
DB2	File system Raw device	Offline, Online	No	DB2 backup using Tivoli Storage Manager API	AIX, HP, NT, SINIX, Solaris
MS SQL Server	File system	Offline, Online	No	TDP for MS SQL server	NT
ADABAS-D	Raw device	Offline, Online	No	ADINT/ Tivoli Storage Manager	AIX, NT, Solaris

20.3 Tivoli Data Protection for R/3

Tivoli Data Protection for R/3 lets you manage backup storage and processing independently from normal SAP R/3 operations. It uses the SAP BACKINT interface to backup and restore a SAP R/3 Oracle database. It allows system administrators to follow SAP procedures and use the integrated SAP utilities for backup and restore. These utilities are SAPDBA, BRBACKUP, BRARCHIVE and

BRRESTORE. Other SAP files can be backed up using Tivoli Storage Manager standard techniques for file backup and restore.

Tivoli Data Protection for R/3 uses the Tivoli Storage Manager API client to communicate between SAP R/3 and the Tivoli Storage Manager server. It has two main modules:

1. Tivoli Data Protection for SAP R/3 PRO: This module deals with the BACKINT interface. It:
    - Receives and sends commands and messages with lists of database objects over the BACKINT interface
    - Reads and interprets the parameters that are specified in the profile
    - Sets up the proper environment and controls the AGENTs
2. Tivoli Data Protection for SAP R/3 AGENT: This module deals with (and contains calls to) the Tivoli Storage Manager API client. It :
    - Reads and writes database objects from and to disk (backup/restore)
    - Transfers data to and from the Tivoli Storage Manager API client. The Tivoli Storage Manager API client in turn connects with the Tivoli Storage Manager servers.

The two components communicate using an internal protocol and can execute on different CPUs. This structure allows parallel backup of multiple nodes of a parallel database.

20.3.1 Alternate Tivoli Storage Manager Servers

For disaster recovery or performance reasons, backup/restore can be routed to more than one Tivoli Storage Manager server. For example, you could store a disaster recovery backup on a remote Tivoli Storage Manager server system at specified intervals.

You can increase throughput by defining multiple backup servers. When you define multiple backup servers, Tivoli Data Protection for R/3 transfers database objects among several Tivoli Storage Manager servers simultaneously.

20.3.2 Performance Enhancements

Tivoli Data Protection for R/3 is multi-threaded and contains the following performance enhancements:

- Multi-threading code using large internal buffers. This allows SMP exploitation.
- Fast null block compression — Designed and typically used for database files since they usually contain large portions of null blocks.
- Multiplexing — Multiplexes several files into a joint data stream allowing for performance optimization depending on the hardware environment and compressibility of the data files.
- User-adjustable block size for reading from disk.
- User-adjustable block size for sending data to Tivoli Storage Manager.

20.3.3 BACKINT File Manager

BACKINT File Manager is an add-on utility of Tivoli Data Protection for R/3. It simplifies the Tivoli Data Protection for R/3 inquire, restore and delete operations. The BACKINT File Manager user interface consists of a text window that displays all the backup IDs found on all Tivoli Storage Manager servers along with all the files that belong to a particular backup ID. You can select individual backup IDs or multiple files to restore or delete.

20.3.4 Administration Tools

The Administration Tools for Tivoli Data Protection for R/3 are an extra-charge optional product. They provide a browser interface to three main tools:

- System Configuration Tool lets you customize the SAP backup profile, the Tivoli Data Protection for SAP R/3 profile, and all the necessary Tivoli Storage Manager files.
- The Performance Monitor displays Tivoli Data Protection for SAP R/3 performance information while Tivoli Data Protection for R/3 is performing a backup or restore operation. You can also display saved performance statistics even when Tivoli Data Protection for R/3 is not active.
- The Operations Monitor provides a centralized overview of backup status information for all SAP R/3 systems registered with the central Administration Tools server. The overview panel shows summaries of the backup status of an SAP R/3 system and the state of each current backup.

20.4 IBM Content Manager CommonStore for SAP

IBM Content Manager CommonStore for SAP (formerly known as EDMSuite CommonStore) is a certified e-business solution for SAP R/3. It provides electronic archive, management and distribution of R/3 data.CommonStore for SAP uses the services of Tivoli Storage Manager to support archive, long-term storage and fast retrieval of R/3 and other data. This reduces the overall size of the R/3 databases, improving both normal application performance as well as administrative functions such as backup and restore.

Distribution of documents through the enterprise with CommonStore for SAP is in electronic form, over either the Internet or intranet. It eliminates the need to maintain hard-to-search and maintain hardcopy archives. R/3 users can access required information at the click of a button, using the familiar presentation client interface, since CommonStore is completely integrated into the R/3 environment.

Business processes can be improved using online documents, and electronic workflow management allows for the automation of routine tasks and document distribution. Forms management can also be integrated with the use of the document.

CommonStore uses SAP’s own ArchiveLink™ specification and supports all types of data and documents defined by this. These include:

- Reorganization data. This is data from R/3 databases which is not required to be kept online, but yet must still be easily retrievable. Archiving this data from the production databases frees up valuable disk space.

- Incoming documents such as letters and faxes. These are scanned in and electronically stored.
- Outgoing documents created by SAP R/3, such as invoices and statements.
- Print lists from SAP R/3 reports, such as inventories and balance sheets.
- Externally generated files, such as spreadsheets or word processing documents.

CommonStore for SAP is available on a variety of different operating system platforms. More information is available at the Internet Web site:

`http://www.ibm.com/software/data/commonstore/`

Chapter 21. Tivoli Storage Manager and SQL Server

The most recent version of this material is maintained on the ITSO Internet Web pages at:

`http://www.redbooks.ibm.com/solutions/systemsmanagement/tsm/c_sql.html`

21.1 SQL Server native backup utilities

Microsoft SQL Server is very similar to Sybase in the type of backup utilities that it provides. The built-in backup utilities provide full online database and associated transaction log backups, individual transaction log backups, full database and transaction log restores, and individual transaction log restores. The command versions of the backup/restore utilities are called dump and load (similar to Sybase).

21.2 Using Tivoli Storage Manager to backup SQL Server

Tivoli Storage Manager integrates with the backup/restore function of Microsoft SQL Server using Tivoli Data Protection for Microsoft SQL Server (TDP for Microsoft SQL). This optional product issues dump and load statements to Microsoft SQL Server using a named pipe option which allows Tivoli Storage Manager to be the receiver of the backup data. TDP for Microsoft SQL uses the Microsoft SQL-DB Library to issue commands to SQL Server. It sends the backup data it receives from SQL Server to a Tivoli Storage Manager server using the Tivoli Storage Manager API. Figure 97 shows the data flow using TDP for SQL Server.

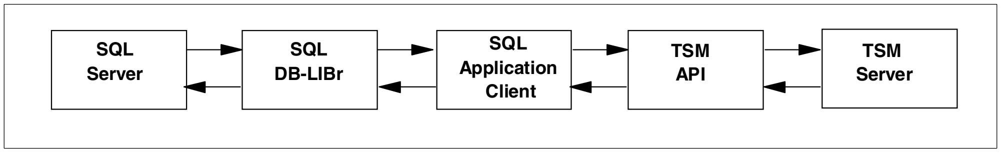


Figure 97. Overview of TDP for MS SQL Server operating environment

TDP for Microsoft SQL Server provides online backup of Microsoft SQL Server databases to Tivoli Storage Manager server storage. Offline backups can be achieved with the standard Tivoli Storage Manager backup/archive client. TDP for Microsoft SQL is cluster aware.

TDP for Microsoft SQL stores all objects as backup objects with unique names, which never expire in Tivoli Storage Manager. A delete function is therefore provided for removing old or unwanted backup objects from Tivoli Storage Manager. You should use the policy recommendation provided in *Tivoli Data Protection for Microsoft SQL Server's Installation Guide*, SH26-4111.

TDP for Microsoft SQL comes with both command line and graphical user interface (GUI) for configuration, deletion of backups, and backup and restore of SQL databases and transaction logs. Backup and restore of SQL databases and transaction logs can be performed via the GUI interface from any SQL server to any Tivoli Storage Manager server with drag-and-drop ease.

Chapter 22. Lotus Notes R4 and Tivoli Storage Manager

The most recent version of this material is maintained on the ITSO Internet Web pages at:

```
http://www.redbooks.ibm.com/solutions/systemsmanagement/tsm/c_notes.html
```

Pointer to "how-to" information:

An ITSO "how-to" guide is available as a companion to this conceptual information. It provides material describing how to implement and operate this feature or function. It is available as a hardcopy redbook, *Using Tivoli Storage Manager to Back Up Lotus Notes*, SG24-4534.

In this chapter, we give a brief introduction to Lotus Notes R4 (hereafter called Notes R4), its components and interfaces, and Notes as a database system. We discuss the importance of storage management and its requirements to a backup solution.

Tivoli Storage Manager provides a backup solution for a heterogeneous Notes R4 environment, including the backup/archive client and the Tivoli Storage Manager for Lotus Notes R4 agent. We show that although each component on its own has limitations in a Notes environment, together they provide a complete backup solution and fulfill the requirements for Notes storage management.

22.1 What is Lotus Notes R4?

Notes R4 is an example of a growing genre of software called *groupware.* Groupware, by definition, means that the data for an application is used by members of a group. Its infrastructure comprises technologies for communication (electronic mail, messaging), collaboration (electronic discussions, shared databases, forums), and coordination (workflow automation).

Notes R4 combines an advanced client/server application development environment, a document database, and a sophisticated messaging system. With Notes R4, people can work together regardless of software or hardware platform; and across technical, organizational, and geographical boundaries. It allows users to communicate securely over a local area network (LAN) or through telecommunication, and create or access documents residing on a shared computer. Notes-based information can be shared across any distance, at any time.

22.1.1 Notes R4 components and platforms

Notes R4 is a client/server application that runs on a variety of workstation operating system platforms. There are two product components:

1. Notes R4 server

   The Notes R4 server provides services to Notes R4 workstation users and other Notes R4 servers, including storage and replication of shared databases and mail routing. With Notes Release 4.6.6 (the latest supported level for TDP for Lotus Notes R4 at the time of writing), you can run the Notes R4 server on the following platforms:

- Windows NT and Windowss 95/98
- OS/390
- OS/2
- NetWare
- UNIX: IBM AIX, HP-UX, and Sun Solaris
- AS/400 systems

2. Notes R4 client

   The Notes client communicates with one or more Notes servers, providing the interface that allows a Notes user to access shared databases and to read and send mail. With Notes R4.6.7, you can run the Notes client on the following platforms:

   - Windows NT and Windows 95/98
   - UNIX: The IBMR4 client is required on user workstations that will access databases on a Notes server. However, the Notes client provides more than just client access to server databases. It is a single-user version of a Notes server, providing a very complete subset of the server functions on the user's client workstation. Notes users can work with databases on a Notes server that are shared with other users or have Notes databases on their own client workstation. The local databases on their client workstation look and feel exactly the same as databases on a server. The only limitation is that the database can be accessed only at the user's client workstation, not by other users in the network.

22.1.2 Notes R4 user interface

Notes R4 provides a graphical user interface (GUI) also called *Notes workspace*, as shown in Figure 98. The NotesR4 workspace is used to access databases or applications. The Notes R4 workspace consists of seven tabbed desktop windows, or folders: six of them contain icons that represent databases. The replicator window enables the user to start and monitor database replications between a Notes R4 server and the local Notes workstation.

Users can open databases by clicking on the tab of a folder and then double-clicking on a database icon. Icons can represent databases or applications on either the user's local workstation or a remote Notes server.

Notes databases and Notes applications are synonymous terms. The Notes R4 workspace also has a set of pull-down menus for working with databases and SmartIcons that provide a fast path to many everyday functions. The Notes R4 workspace has the same look and feel across all of the supported client platforms.

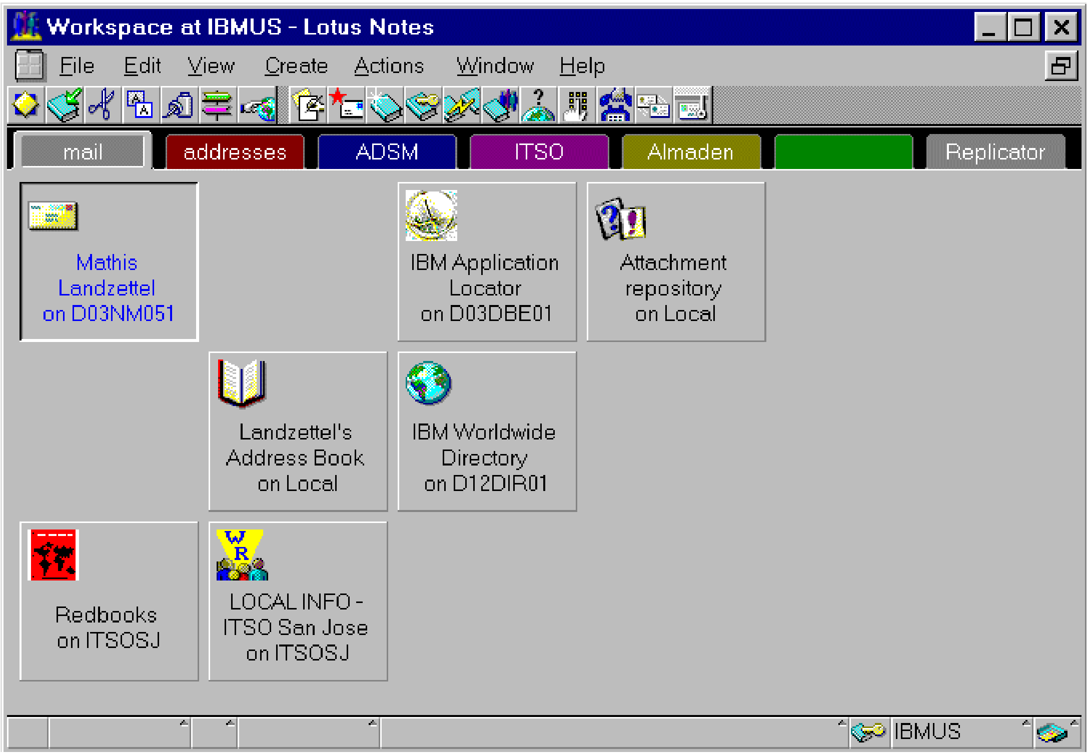

Workspace at IBMUS - Lotus Notes
File
Edit
View
Create
Actions
Window
Help
mail
addresses
ADSM
ITSO
Almaden
Replicator
Mathis Landzettel on D03NM051
IBM Application Locator on D03DBE01
Attachment repository on Local
Landzettel's Address Book on Local
IBM Worldwide Directory on D12DIR01
Redbooks on ITSOSJ
LOCAL INFO - ITSO San Jose on ITSOSJ
IBMUS

Figure 98. Notes R4 workspace

22.1.3 Notes R4 administration

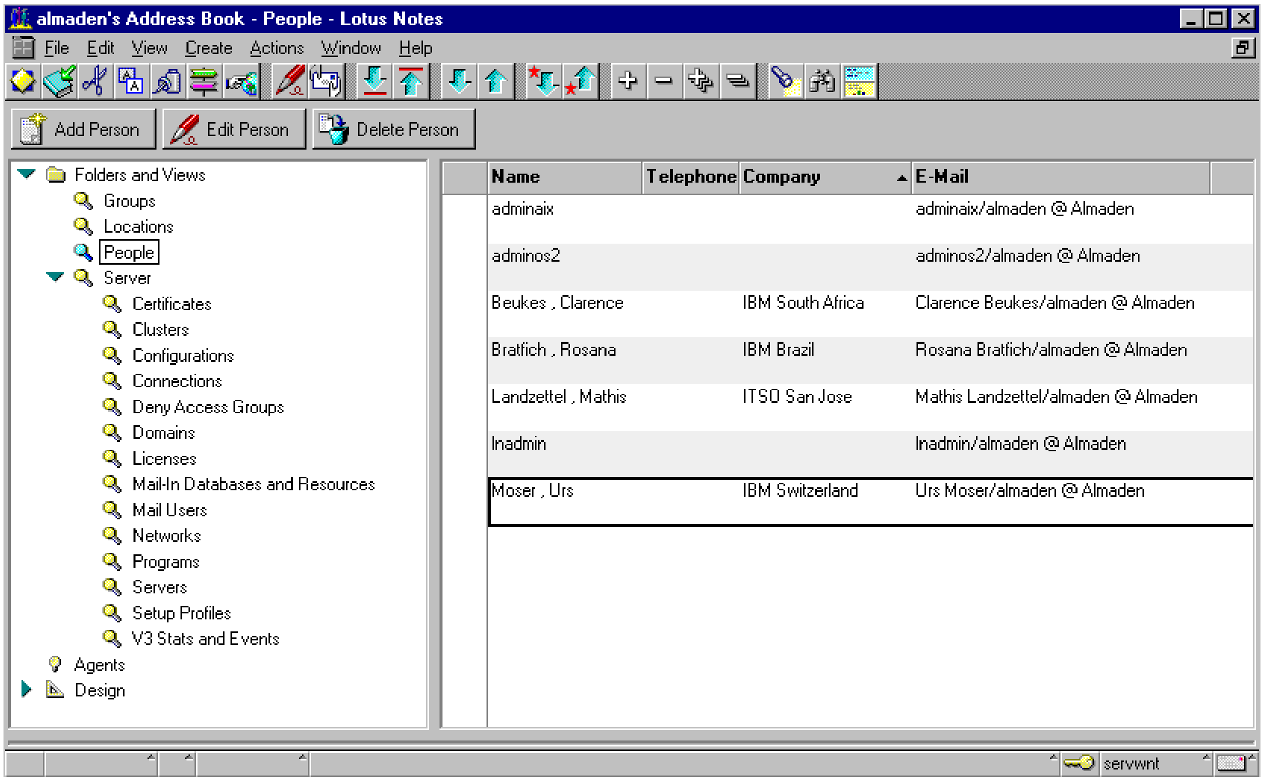
Like any other system, Notes requires administration. Notes R4 provides two main interfaces for administration: the Notes R4 server console and the Notes client workspace.

When a Notes R4 server is started, a full-screen command line console is presented. The console displays the server activities, such as scheduled macros and replication. The server console is also the interface for administrators to perform tasks such as loading additional Notes programs, querying server statistics, and setting certain server options. The server console is a window on the physical server screen. A remote console function is also provided through the Notes R4 workspace. It enables remote server administration by suitably authorized users.

The following is an example of a typical console display:

```
Lotus Domino r Server, Release 4.5.2 (Intl), 26 June 1997
Copyright c 1985-1997, Lotus Development Corporation, All Rights Reserved

04/02/98 12:58:32 PM  Database Replicator started
04/02/98 12:58:37 PM  Mail Router started for domain ALMADEN
04/02/98 12:58:37 PM  Router: Internet SMTP host malawi in domain almaden.ibm.com
04/02/98 12:58:42 PM  Index update process started
04/02/98 12:58:47 PM  Stats agent shutdown
04/02/98 12:58:52 PM  Agent Manager started
04/02/98 12:58:55 PM  AMgr: Executive '1' started
04/02/98 12:58:57 PM  servwnt/almaden is the Administration Server of the Public
Address Book.
04/02/98 12:58:58 PM  Administration Process started
04/02/98 12:59:02 PM  Schedule Manager started
04/02/98 12:59:02 PM  SchedMgr: Validating Schedule Database
04/02/98 12:59:03 PM  SchedMgr: Done validating Schedule Database
04/02/98 12:59:07 PM  Calendar Connector started
04/02/98 12:59:13 PM  HTTP Web Server started
04/02/98 12:59:18 PM  Database Server started
```

Although the server console is the interface for basic administration, most Notes administration is performed by using a special Notes R4 database called the *Name & Address database*, from the Notes R4 workspace. The Name & Address database, with a file name of names.nsf, is created on every Notes R4 server and client when the servers and clients are installed. On a Notes R4 server, the Name & Address databases (also called *books*) are public, and contain information about all servers and users within the network. On a Notes R4 client, the Name & Address database is private and contains information pertinent only to that client workstation. Figure 99 shows the Notes R4 server Name & Address database, called *Almaden's Address Book*. The selected view is *People,* which lists all registered users in the Notes domain Almaden. A Notes domain is defined as a collection of users, servers, and groups that share a common Name & Address book within a Notes environment.



Name	Telephone	Company	E-Mail
adminaix			adminaix/almaden @ Almaden
adminos2			adminos2/almaden @ Almaden
Beukes , Clarence		IBM South Africa	Clarence Beukes/almaden @ Almaden
Bratfich , Rosana		IBM Brazil	Rosana Bratfich/almaden @ Almaden
Landzettel , Mathis		ITSO San Jose	Mathis Landzettel/almaden @ Almaden
lnadmin			lnadmin/almaden @ Almaden
Moser , Urs		IBM Switzerland	Urs Moser/almaden @ Almaden

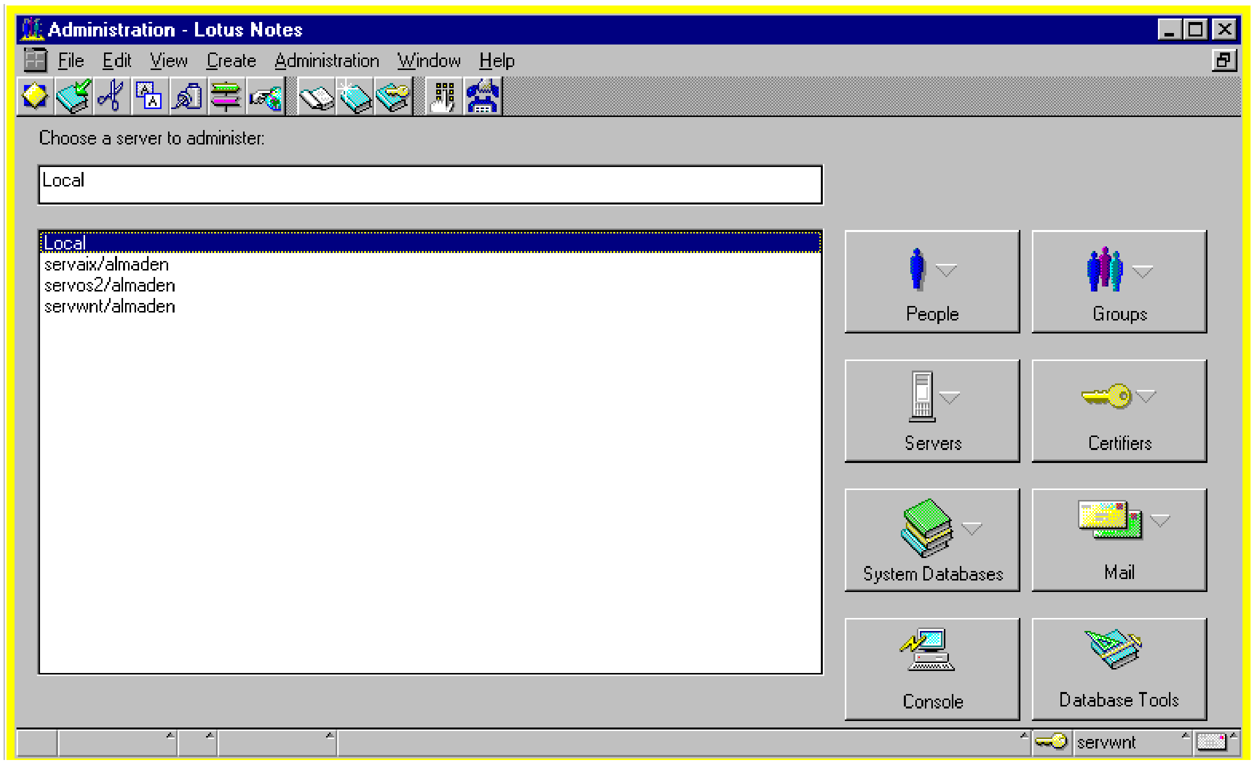
Figure 99. Public Name & Address book

On a Notes R4 server, the Name & Address database is probably the most powerful directory services tool and server management tool for an administrator. As a directory service tool, the public Name & Address database provides a directory of all Notes users, servers in a domain, group names for mailing lists, and foreign domains. Servers within a domain have a common Name & Address database that is replicated across all servers in the domain.

As a server management tool, the Name & Address database is used to control server operations. It contains instructions on how servers can communicate with other servers, and which tools or programs are run. It is the main Notes scheduling tool to schedule operations such as establishing server-to-server connections for replication.

Notes R4 administrators additionally use the Notes R4 workspace to administer a Notes server. If you select **File —> Tools —> Server Administration,** a dedicated administration window appears. From there, administrators can maintain all Notes R4 servers, register new users or servers in the domain, and access the different views of the Name & Address database.

A typical administration window on a Notes R4 workspace is shown in Figure 100.



Choose a server to administer:

Local

Local
servaix/almaden
servos2/almaden
servwnt/almaden

Figure 100. Administration window on Notes R4 workspace

22.1.4 Notes data

A Notes database is the basic component of a Notes R4 application. It is the repository where users create, update, store, and track documents in various formats. The document-oriented information within the files is unstructured and can contain many types of data: text, image, audio, and video. Shared databases reside on one or more Notes R4 servers and can be accessed by multiple users. A local database is resident on a user's client workstation and is accessible only at that workstation. A Notes database created on a Windows client, for example, has the same format as a database created on an OS/2, NetWare, or UNIX Notes server. Therefore, Notes databases are portable among various Notes servers and clients throughout an enterprise.

A Notes database is stored on a server or client workstation as a single Notes structured file with an .nsf file extension. A Notes database is a single, self-contained entity as far as the workstation operating system is concerned. A workstation operating system such as AIX has no knowledge of the structure or contents of a Notes database. The lack of knowledge is beneficial in terms of portability but presents an interesting storage management challenge. Notes databases can become very large files, often growing to hundreds of megabytes in size.

Beside databases, Notes R4 includes a number of other non-database files. These are not Notes databases, but are normal operating system files. They include initialization files, ID files, configuration and others. The point here is that a Notes R4 environment is more than just the databases and that these non-database files must also be considered in the backup strategy.

22.2 Tivoli Storage Manager backup/archive client and Notes R4

The Tivoli Storage Manager backup/archive client is designed for file system data. Tivoli Storage Manager can backup all non-database Notes data on both Notes R4 servers and clients. Tivoli Storage Manager clients use standard operating system functions to access files within file systems, but they do not understand any logical structure that might exist within a file. This limitation affects how Notes and other database systems are backed up. Each database appears as an individual NSF file in the server or client file systems. A Tivoli Storage Manager backup/archive client running on a Notes R4 server or client can back up restore, archive, and retrieve entire Notes databases. It cannot back up smaller increments.

Tivoli Storage Manager clients can be installed wherever there are Notes databases that require backing up. However, that approach could potentially lead to large numbers of duplicate database backup copies if Notes replication is also being used. A more sensible approach is to implement Tivoli Storage Manager clients on Notes R4 servers only. If possible, identify those databases on the servers that are replicas from other servers and exclude them from backup. (This approach assumes that backups of those databases have already been performed at the originating database server.)

Other than the issues of size and replication, using a Tivoli Storage Manager backup/archive client to back up Notes databases is straightforward. Each database is a self-contained NSF file that is backed up and restored without any problem. Tivoli Storage Manager restores a database in its entirety because it is just a file for Tivoli Storage Manager. If a database is deleted or corrupted, it is a simple task for Tivoli Storage Manager to restore the most recent or any backup copy version of this database from the Tivoli Storage Manager server to the Notes R4 server or client.

The Tivoli Storage Manager backup/archive client, however, does not meet all requirements for an ideal storage management solution in a Notes R4 environment, hence the need for the application agent, Tivoli Data Protection for Lotus Notes.

22.3 Tivoli Data Protection for Lotus Notes R4

Tivoli Data Protection (TDP) for Lotus NotesR4 is a separate Tivoli Storage Manager application client available for Windows NT and IBM AIX. For OS/2, the Notes R4 backup agent was provided at no charge in older releases of the standard OS/2 Tivoli Storage Manager backup/archive client. Note that starting from V3.7 of the OS/2 Tivoli Storage Manager client, the OS/2 Notes agent is no longer shipped. For simplicity, we will specify TDP for Lotus Notes or TDP for Notes from now on, although the functions described refer equally to the OS/2 Notes agent as well.

Tivoli Data Protection for Lotus Notes provides an integrated solution for performing document-level incremental backup and restore operations on a Notes database. It therefore provides functions complementary to the standard

Tivoli Storage Manager backup/archive client. Table 11 compares the functions provided by the Tivoli Storage Manager backup/archive client and TDP for Notes.

Table 11. Backup/archive client and TDP for Notes functions

Function	**backup/archive client**	**TDP for Notes**
Non-database backup/restore	Yes	No
Full database backup/restore	Yes	Yes
Incremental database backup	No	Yes
Restore individual documents	No	Yes
Restore deleted documents	No	Yes

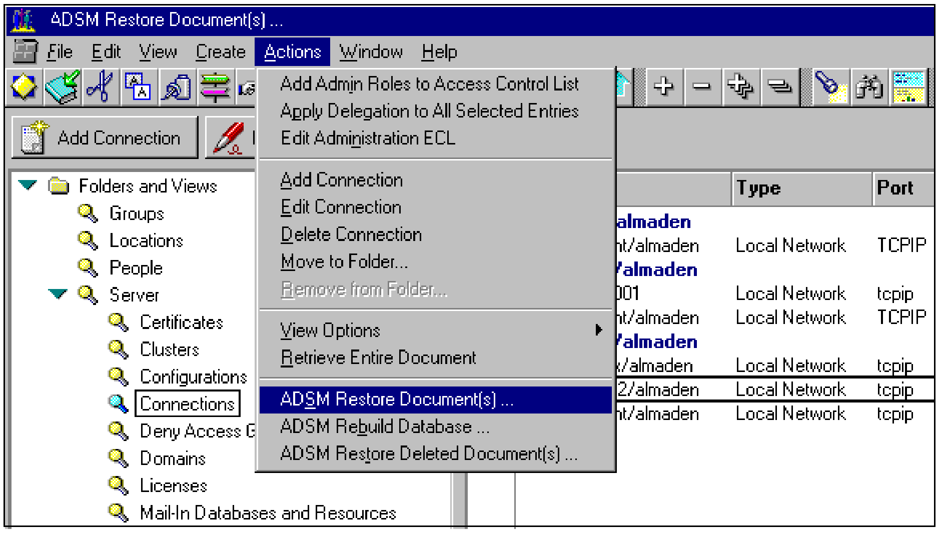
TDP for Notes cannot be used to back up or restore any other type of data, such as Notes ID files, or notes.ini, or any other system configuration files. Those files are backed up by the Tivoli Storage Manager backup/archive client.

TDP for Notes provides the ability to back up Notes databases at the individual Notes document level. TDP for Notes is integrated within Notes and recognizes the logical structure of a Notes database.

TDP for Notes provides commands to:

- Perform an incremental backup of Notes databases, backing up only those documents that have changed since the previous incremental backup.
- Restore to a database individual documents that have been previously backed up by the Notes agent.
- Restore individual documents that have been deleted from a Notes database.
- Restore an entire Notes database.
- Perform an incremental restore function by merging changed documents into a database.

TDP for Notes provides a command line interface for performing backups and restores. TDP for Notes commands can be issued either from a command prompt or from a Notes server console. You can invoke the Notes agent GUI from the Notes workspace. Figure 101 shows the TDP for Notes functions embedded in the Notes workspace.



	Type	Port
almaden		
ht/almaden	Local Network	TCPIP
/almaden		
001	Local Network	tcpip
ht/almaden	Local Network	TCPIP
/almaden		
/almaden	Local Network	tcpip
2/almaden	Local Network	tcpip
ht/almaden	Local Network	tcpip

Figure 101. TDP for Notes GUI

TDP for Notes uses the Notes R4 API to communicate with the Tivoli Storage Manager API. The Notes F4 API enables TDP for Notes to understand the structure within a Notes database and therefore to back up and restore on the document level. The Tivoli Storage Manager API, which has a C language interface, can be regarded as a special type of Tivoli Storage Manager client. It is designed to provide an interface between a workstation application and a Tivoli Storage Manager server. The Tivoli Storage Manager API is configured in a way similar to the standard Tivoli Storage Manager backup/archive client. A Tivoli Storage Manager server views an application using the API as a separate, self-contained workstation client. Figure 102 illustrates the logical components of the Notes agent.

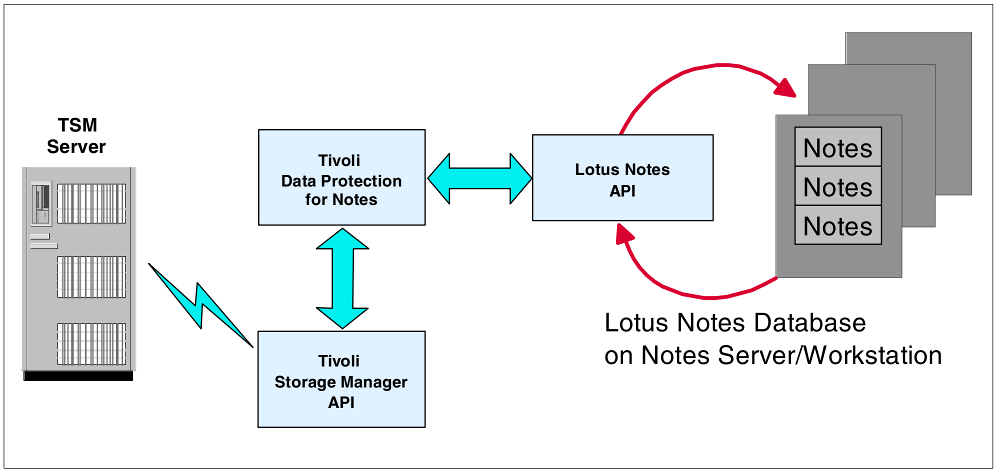

TSM Server
Tivoli Data Protection for Notes
Lotus Notes API
Tivoli Storage Manager API
Notes
Notes
Notes
Lotus Notes Database on Notes Server/Workstation

Figure 102. Logical components of TDP for Notes

Chapter 23. Lotus Domino R5 and Tivoli Storage Manager

The most recent version of this material is maintained on the ITSO Internet Web pages at:

`http://www.redbooks.ibm.com/solutions/systemsmanagement/tsm/c_notes.html`

In this chapter, we give a brief introduction to Lotus Domino R5 (hereafter referred to as Lotus Domino), its components, interfaces and Lotus Domino as a database system. We will discuss the importance of storage management and its requirements for a backup solution.

Tivoli Storage Manager provides a backup solution for a heterogeneous Domino environment, which includes the backup/archive client and the Tivoli Data Protection (TDP) agent for Lotus Domino. Together, these provide a complete backup solution to fulfill the requirements of Lotus Domino storage management.

23.1 What is Lotus Domino R5?

Lotus Domino includes the Notes client and the Domino server for messaging, collaboration, workflow automation, Internet and intranet applications.

Lotus Domino comprises an advanced client/server environment, a document database, and a sophisticated messaging system. With Lotus Domino, users can work together regardless of software or hardware platform, and across technical, organizational, and geographical boundaries. It allows users to communicate securely over a local area network (LAN) or through telecommunication. Lotus Domino enables users to create and access documents across any distance, at any time.

23.1.1 Lotus Domino components and platforms

Lotus Domino is a client/server application that runs on a variety of operating system platforms. There are two product components:

1. Lotus Domino (server)

   There are three types of Lotus Domino servers:

   - **Lotus Domino R5 Mail Server:** The Lotus Domino Mail Server combines full support for the latest internet mail standards with Domino's industry-leading messaging capabilities.
   - **Lotus Domino R5 Application Server:** The Lotus Domino Application Server is an open, secure platform optimized to deliver collaborative Web applications that integrate your enterprise systems with rapidly changing business process.
   - **Lotus Domino R5 Enterprise Server:** The Lotus Domino Enterprise server delivers all the functionality of the Domino Mail and Application servers reinforced with clustering for the high availability and reliability required by mission-critical applications.

   The Domino server provides services to Notes client users and other Domino servers, including storage and replication of shared databases and mail routing. The Domino server runs on the following platforms:

   - Windows NT

- OS/390
- OS/2
- UNIX: IBM AIX, HP-UX, Linux and Sun Solaris

2. Lotus Notes (client)

   There are three types of Lotus Notes clients:

   - **Lotus Notes:** The Lotus Notes client provides the interface between the user and Lotus Domino server. It includes all features and functionality which is required by the end user. Users cannot perform system administration or application development(design) using this client.
   - **Lotus Domino Administrator:** The Lotus Domino Administrator client is developed exclusively for Lotus Domino system administration. It gives a complete range of system administration tools in order to make administration easy with less effort.
   - **Lotus Domino Designer:** The Lotus Domino Designer is developed for Lotus Domino Application Developers. It is used to create and modify the Domino applications. This client provides a wide range of development tools which allows developers to give their best software solutions.

   The Notes client(s) communicate with one or more Notes servers, providing the interface that allows a Notes user to access/administer/develop Domino Applications (databases). With Notes R5, you can run Notes client on the following platforms:

   - Windows NT & Windows 95/98
   - Apple Macintosh

Notes databases can reside on the Domino server or on the individual Notes client. Databases which reside on the client can be accessed only by that user and administration of these database is much simpler than administering the databases on the Domino server.

23.1.2 Lotus Notes user interface

Lotus Notes R5 provides a new graphical user interface (GUI). It is designed for both intranet and Internet users. The default initial screen is the "Welcome Page", which can be customized to users’ own personal preferences.

It also gives an option for Notes R4 users to use the Notes R4 GUI, so that, Notes R4 users will have a familiar environment when the software is upgraded to Notes The screenshot of Lotus Notes R5 Welcome Page is shown in Figure 103.

Notes databases and Notes applications are synonymous terms. The Notes R5 GUI also has a set of pull-down menus for working with databases and SmartIcons that provide a fast path to many everyday functions. The Notes R5 GUI has the same look and feel across all of the supported client platforms.

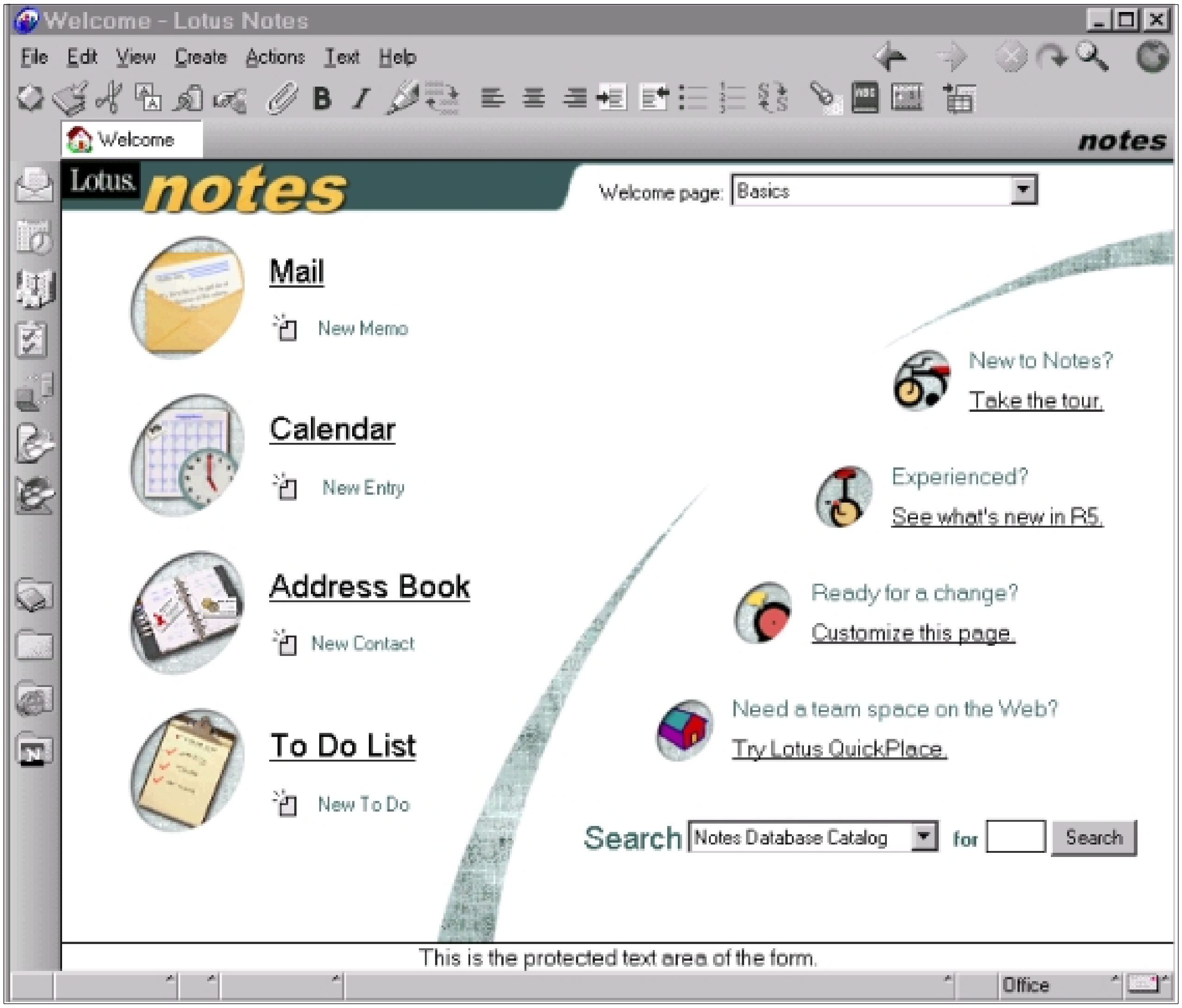


Figure 103. Notes Welcome Page

23.1.3 Lotus Domino Administrator

Like any other system, Domino requires administration. Domino provides two main interfaces for administration: the Domino server console and the Lotus Domino Administrator client.

When a Domino server is started, a full-screen command line console is presented. The console displays the server activities, such as scheduled macros and replication. The server console is also the interface for administrators to perform tasks such as loading additional Notes programs, querying server statistics, and setting certain server options. The server console is a window on the physical server's screen. A remote console function is also provided through the Lotus Domino Administrator which enables remote server administration by suitably authorized users. The following is an example of a typical console display:

```
Lotus Domino (r) Server, Build V5010707 (Intl), 07 July 1999
Copyright (c) 1985-1999, Lotus Development Corporation, All Rights Reserved

Restart Analysis: 100%
02/23/2000 12:39:33 PM  Recovery Manager: Restart Recovery complete. (0/0
databases needed full/partial recovery)
The ID file being used is: /victoria/victoria.id
Enter password (press the Esc key to abort):

02/23/2000 12:39:40 PM  Mail Router started for domain ALMADEN
02/23/2000 12:39:40 PM  Router: Internet SMTP host victoria in domain
almaden.ibm.com
02/23/2000 12:39:45 PM  Database Replicator started
02/23/2000 12:39:50 PM  Index update process started
02/23/2000 12:39:55 PM  Agent Manager started
02/23/2000 12:39:56 PM  AMgr: Executive '1' started
02/23/2000 12:40:00 PM  lochness/Almaden is the Administration Server of the
Domino Directory.
02/23/2000 12:40:00 PM  Administration Process started
02/23/2000 12:40:05 PM  Calendar Connector started
02/23/2000 12:40:10 PM  Schedule Manager started
02/23/2000 12:40:10 PM  SchedMgr: Validating Schedule Database
02/23/2000 12:40:10 PM  SchedMgr: Done validating Schedule Database
02/23/2000 12:40:15 PM  Event Dispatcher started
02/23/2000 12:40:20 PM  Stats agent started
02/23/2000 12:40:24 PM  Database Server started
>
```

Lotus Domino R5 provides a special client for Lotus Domino System Administration called the Lotus Domino Administrator. This client is used for remote system administration, where an administrator can perform all administrative operations remotely, without even going to the physical server.

Figure 104 shows the Domino Administrator with the people and groups view open.

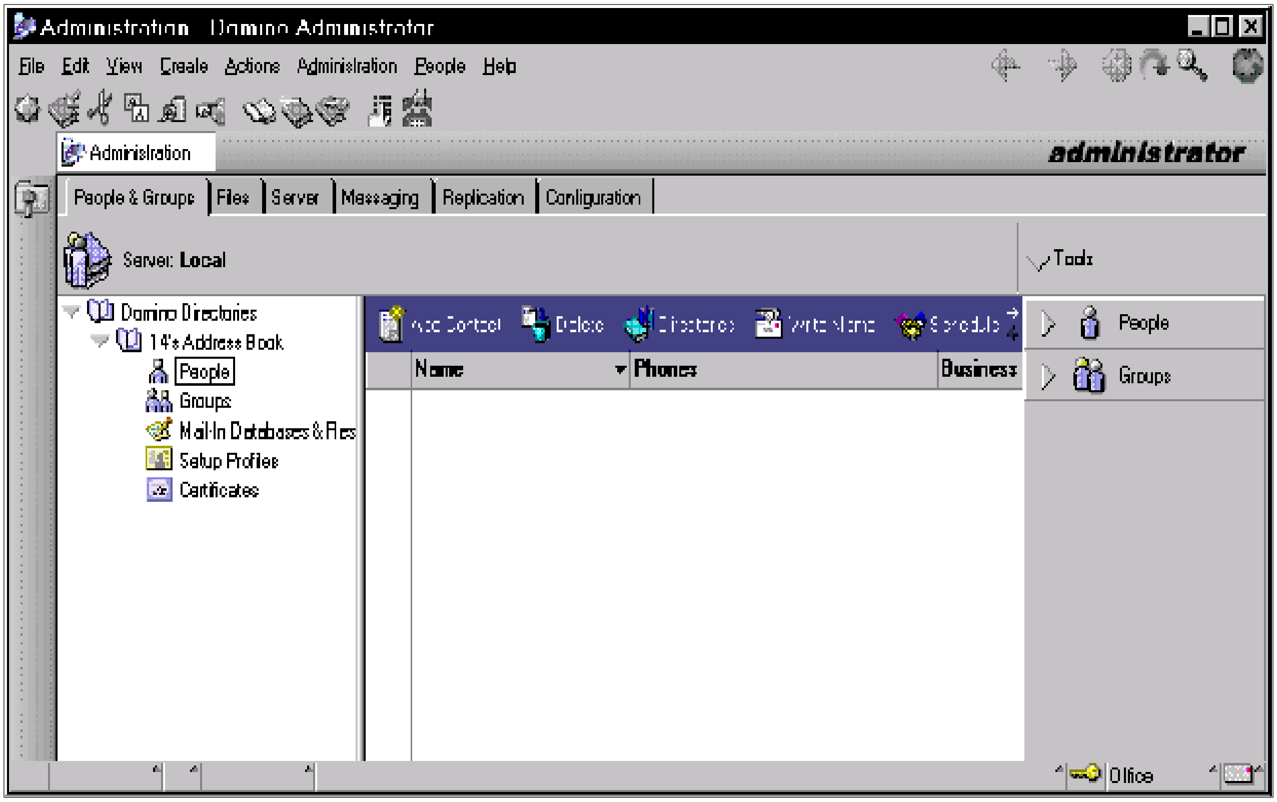
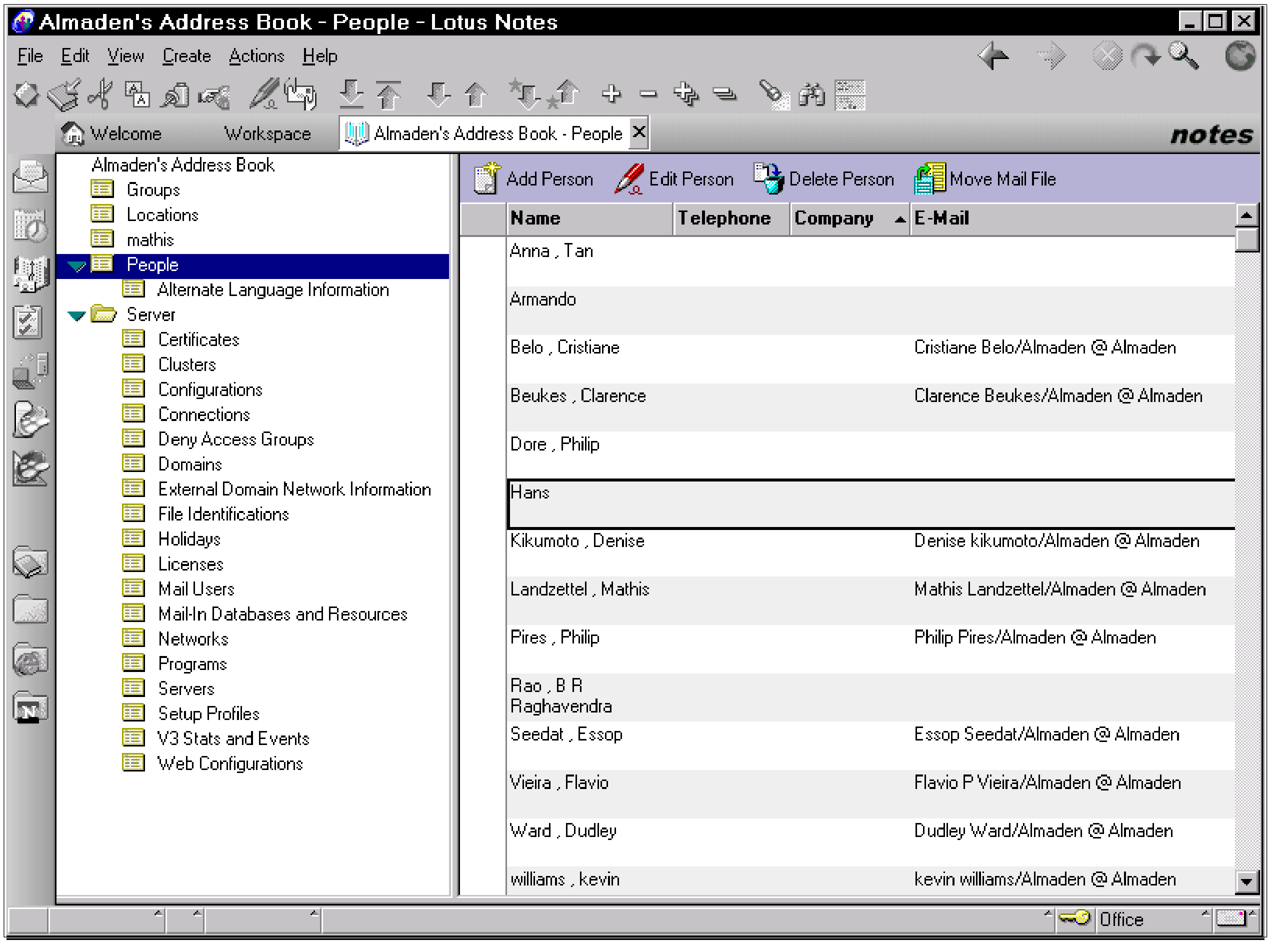


Figure 104. Lotus Domino Administrator

Although the server console and Lotus Domino Administrator is the interface for basic administration, most Domino administration is performed by using a special Notes database called the *Domino Directory (*which in Notes R4 was called the *Name & Address database)*, from the Lotus Notes client. The Domino Directory database, with a file name of names.nsf, is created on every Domino server and client when the servers and clients are installed. On a Notes server, the Domino Directory is public, and contain information about all servers and users within the network. On a Notes client, the Domino Directory database is private and contains information pertinent only to that client. Figure 105 shows the Domino server Domino Directory database, called *Almaden's Address Book*. The selected view is *People,* which lists all registered users in the Notes domain Almaden. A Notes domain is defined as a collection of users, servers, and groups that share a common Domino Directory within a Notes environment.



Name	Telephone	Company	E-Mail
Anna , Tan			
Armando			
Belo , Cristiane			Cristiane Belo/Almaden @ Almaden
Beukes , Clarence			Clarence Beukes/Almaden @ Almaden
Dore , Philip			
Hans			
Kikumoto , Denise			Denise kikumoto/Almaden @ Almaden
Landzettel , Mathis			Mathis Landzettel/Almaden @ Almaden
Pires , Philip			Philip Pires/Almaden @ Almaden
Rao , B R Raghavendra			
Seedat , Essop			Essop Seedat/Almaden @ Almaden
Vieira , Flavio			Flavio P Vieira/Almaden @ Almaden
Ward , Dudley			Dudley Ward/Almaden @ Almaden
williams , kevin			kevin williams/Almaden @ Almaden

Figure 105. Domino Directory

On a Domino server, the Domino Directory database is probably the most powerful directory services tool and server management tool for an administrator. As a directory service tool, the Domino Directory database provides a directory of all Notes users, servers in a domain, group names for mailing lists, and foreign domains. Servers within a domain have a common Domino Directory database that is replicated across all servers in the domain.

As a server management tool, the Domino Directory database is used to control server operations. It contains instructions on how servers can communicate with other servers, and which tools or programs are run. It is the main Notes scheduling tool for operations such as establishing server-to-server connections for replication.

23.1.4 Notes data

Notes data consists of both database and non-database files. A Notes database is the basic component of a Notes application. It is the repository where users create, update, store, and track documents in various formats. The document-oriented information within the files is unstructured and can contain many types of data: text, image, audio, and video. Shared databases reside on one or more Domino servers and can be accessed by multiple users. A local database is resident on a user's client and is accessible only at that client. A

Notes database created on a Windows client, for example, has the same format as a database created on an OS/2 or UNIX Domino server. Therefore, Notes databases are portable among various Domino servers and clients throughout an enterprise.

A Notes database is stored on a server or client as a single notes structured file with a .nsf file extension. A Notes database is a single, self-contained entity as far as the client operating system is concerned. Notes databases can become very large files, often growing to hundreds of megabytes in size. A client operating system such as AIX has no knowledge of the internal structure or contents of a Notes database. This lack of knowledge is beneficial in terms of portability but presents an interesting storage management challenge.

Beside databases, Domino includes a number of other non-database files. These are not Notes databases, but are normal operating system files. They include initialization files, ID files, configuration and others. The point here is that a Domino environment is more than just the databases and that these non-database files must also be considered in the backup strategy.

23.1.5 Storage management of Domino

Providing effective storage management services for a Domino system can be a demanding task. All non-database Domino data comes under your general storage management policy: regular backups need to be run against frequently updated data. The challenge with Domino, however, is the storage management of the Notes databases.

Notes databases are complex logical structures, often very large, that appear to traditional storage management tools as single client files. A backup tool that operates only at operating system level will always back up the entire database. Whenever a single document is updated within a database, an incremental backup would catch the entire database, since the modification time stamp of the database has changed. This could lead to an enormous amount of data and backup copies on the storage location.

Notes itself provides a replication function for database backups. Replication is the process of updating databases that reside on different servers and clients within a Domino environment simultaneously. Updates to a database can be reflected on all database copies wherever they physically reside. This update works on a document level. If a database or a document within a database is accidentally deleted, it can be recovered as long as a replication database copy is available elsewhere in the Domino environment. However, replication is not a substitute for an effective backup solution. Replication will duplicate user errors throughout a Domino network. If a critical document or database is erased by accident, replication will, in time erase that same information wherever it is replicated.

Another problem with most backup products is that they do not allow backup of open files. There are several files that cannot be backed up while the Domino server is running, one of which is the most crucial file, the Domino Directory database. These files will not be backed up unless you stop the Domino server first, or make periodic replica copies that can be backed up.

23.2 Tivoli Storage Manager backup/archive client and Lotus Domino

The backup/archive client is designed to back up and restore, archive and retrieve client file system data. The client therefore can back up any non-database Notes data on both Domino server and client. Backup/archive clients use standard operating system functions to access files within file systems, but they do not understand any logical structure that might exist within a file. This limitation affects how Domino and other database systems are backed up. Each database appears as an individual NSF file in the server or client file systems. A backup/archive client running on a Domino server or client can only back up and restore entire Notes databases. It cannot back up smaller increments.

Backup/archive clients can be installed wherever there are Notes databases that require backing up. However, that approach could potentially lead to large numbers of duplicate database backup copies if Notes replication is also being used. A more sensible approach is to implement backup/archive clients on Notes servers only. If possible, identify those databases on the servers that are replicas from other servers and exclude them from backup. This approach assumes that backups of those databases have already been performed at the originating database server.

Other than the issues of size and replication, using the backup/archive client to back up Notes databases is straightforward. Each database is a self-contained NSF file that is backed up and restored without any problem. The backup/archive client restores a database in its entirety because it is just a file for Tivoli Storage Manager. If a database is deleted or corrupted, it is a simple task for Tivoli Storage Manager to restore the most recent or any previous backup version of this database from the Tivoli Storage Manager server to the Domino server or client.

The backup/archive client, however, does not meet all requirements for an ideal storage management solution in a Notes environment, hence the need for the application agent, Tivoli Data Protection for Lotus Domino.

23.3 Tivoli Data Protection for Lotus Domino

Tivoli Data Protection (TDP) for Lotus Domino is a Tivoli Storage Manager application client that backs up and restores Lotus Domino databases and transaction logs. When archival logging is used on the Domino server, it archives the transaction log files and retrieves them as required for a database recovery. Database backups and archived transaction log files are stored on Tivoli Storage Manager storage. TDP for Domino is available on the following platforms:

- Microsoft Windows NT and Windows 2000
- IBM AIX
- Sun Solaris
- IBM S/390

TDP for Lotus Domino helps protect and manage Lotus Domino server data by making it easy to perform the following actions:

- Backup Notes databases online
- Maintain multiple backup versions of Notes databases

- Archive transaction log files when archival logging is in effect
- Restore a Notes database and apply changes since the last backup from the transaction log
- Restore a Notes databases to a specific point in time
- Expire database backups automatically based on version limit and retention period
- Expire archived transaction log files when no longer needed
- Automate scheduled backups

TDP for Lotus Domino communicates with a Tivoli Storage Manager server using the Tivoli Storage Manager application program interface (API). TDP for Lotus Domino communicates with a Domino server using the Lotus Domino API. The logical components of TDP for Lotus Domino are shown in Figure 106.

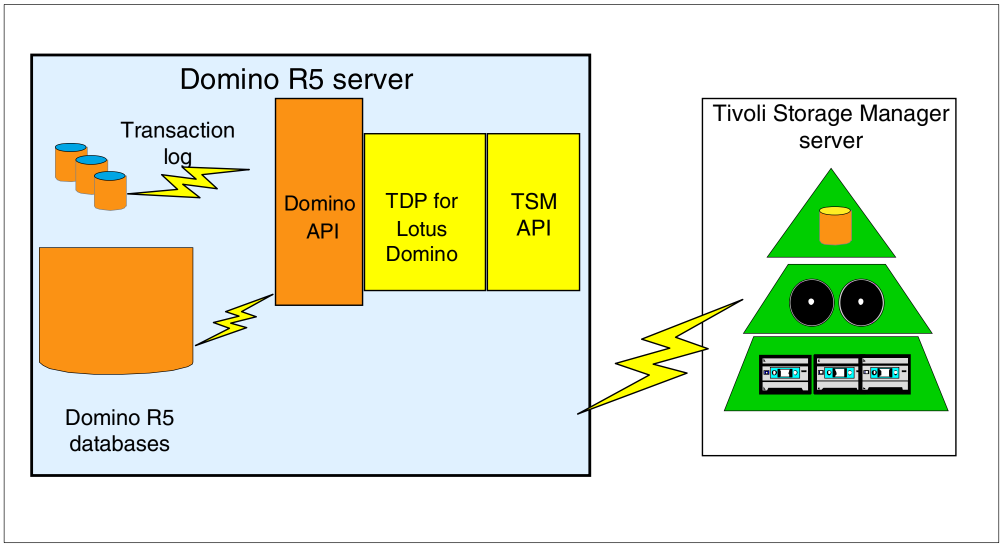


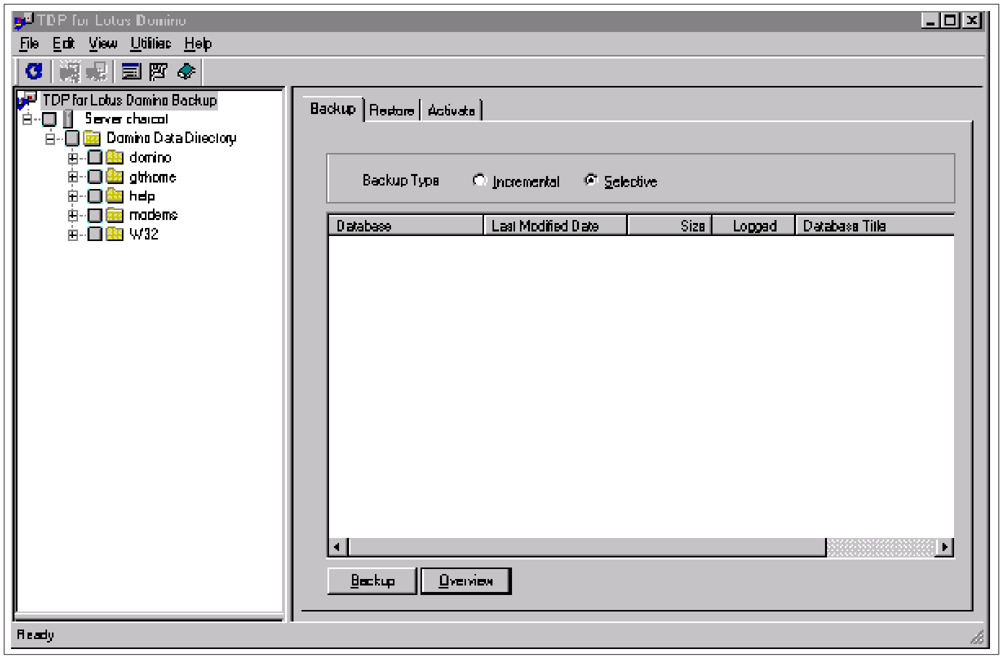
Figure 106. Logical components of TDP for Lotus Domino

The backup and recovery API in Domino R5 provides the capability to perform online full backups of individual databases and archives of the transaction log when archival logging is in effect. A transaction log captures database changes for logged databases, so full database backups are not required as frequently. Updates to a logged database are recorded in the Domino server transaction log. Changes to a database since the last full backup can be applied from the transaction log after the backup is restored from the last full backup.

TDP for Domino cannot be used to back up or restore any other type of data, such as operating system or Domino non-database files. Those files are backed up by the Tivoli Storage Manager backup/archive client.

TDP for Domino provides a command line interface on all platforms and a GUI on Windows NT/2000 only for performing backups and restores.

Figure 107 shows the TDP for Domino GUI interface on a Windows NT system.



- TDP for Lotus Domino Backup
  - Server charcol
    - Domino Data Directory
      - domino
      - gtrhome
      - help
      - modems
      - W32

Backup Type ○ Incremental ● Selective

Database	Last Modified Date	Size	Logged	Database Title

Figure 107. TDP for Domino Notes agent GUI

Chapter 24. Microsoft Exchange

This chapter provides introductory information for Tivoli Data Protection for Microsoft Exchange Server.

Pointer to "how-to" information:

An ITSO "how-to" guide is available as a companion to this conceptual information. It provides material describing how to implement and operate this feature or function. It is available as a hardcopy redbook, Using ADSM to Back Up and Recover Microsoft Exchange Server, SG24-5266.

The most recent version of the guide is maintained on the ITSO Internet Web pages at:

`http://ibm.com/redbooks`

24.1 Tivoli Data Protection for Microsoft Exchange Server

Tivoli Data Protection for Microsoft Exchange Server (hereafter referred to as TDP for Exchange) is a Tivoli Storage Manager application client that provides on-line backups of Microsoft Exchange Server databases to a Tivoli Storage Manager server. TDP for Exchange helps protect and manage Exchange Server data by making it easy to do the following:

- Perform full, copy, differential, and incremental backups of the Microsoft Exchange Directory and Information Store databases
- Restore a full Directory or Information Store database and any number of associated transaction logs
- Delete a Directory or Information Store database backup from Tivoli Storage Manager storage
- Back up the Exchange Server databases to any Tivoli Storage Manager server with drag-and-drop ease
- Set Tivoli Storage Manager options regarding connection information to Tivoli Storage Manager servers
- Launch other Tivoli Storage Manager and related system applications
- Automate scheduled backups
- Automate deletion of old backups

TDP for Exchange communicates with Tivoli Storage Manager using its application program interface (API) and with an Exchange Server using the Exchange API. See Figure 108.

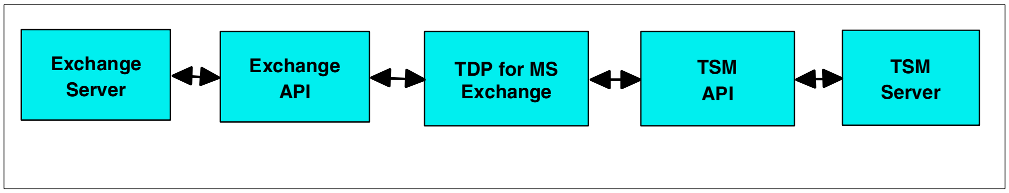


Figure 108. Overview of the TDP for Exchange operating environment

TDP for Exchange must be installed on the same machine as the Exchange Server. This does not need to be the same system which is running the Tivoli Storage Manager server. The Tivoli Storage Manager server can be on any operating system platform, so along as it is Version 3 or higher. TDP for Exchange can compress Exchange data before sending it to the Tivoli Storage Manager server. TDP for Exchange also runs in a Microsoft Cluster Server environment.

24.2 Exchange Application Client functions

This section gives an overview of functions the Exchange Application Client can perform including:

- Backup
- Restore
- Backup delete

In addition, this section identifies security requirements, performance, and backup strategy, and Microsoft Cluster Server considerations.

24.2.1 Exchange Server database backup

A backup creates a copy of an Exchange database with any associated transaction logs on Tivoli Storage Manager storage media.

TDP for Exchange provides selection mechanisms and the logic that is required to back up and restore Exchange data. There are four types of backup provided by TDP for Exchange:

- **Full backup**: A full backup backs up the specified database as well as its associated transaction logs. After the database and logs are backed up, the log files are deleted.
- **Copy backup**: A copy backup is similar to a full backup except that transaction log files are not deleted after the backup. A copy backup can be used to make a full backup of the Exchange Server database without disrupting any backup procedures that use incremental or differential backups.
- **Incremental backup**: An incremental backup only backs up the transaction logs, and then deletes them. Restoration of an Exchange Server database from an incremental backup requires:
  - Restore of the last full backup
  - Restore of any other incremental backups performed between the full backup and this incremental backup
  - Restore of this incremental backup

- **Differential backup**: A differential backup only backs up transaction logs, but does not delete them. If you perform a full backup and then perform only differential backups, the last full backup plus the latest differential backup has all data needed to bring the database back to the most recent state. This type of backup is also called a *cumulative incremental* backup.

  Restoring an Exchange Server database from a differential backup requires the following:

  - Restore of the last full backup
  - Restore of this differential backup, but no other differential backups

Note: When the Exchange Server is installed, circular logging is set as the default and is enabled for both the Directory and Information Store databases. When circular logging is enabled, you cannot use differential or incremental backups. The reason for this is because data loss could occur if the log wrapped before an incremental or differential backup is done. If you choose a backup strategy that involves incremental or differential backups, you must disable circular logging for the Exchange databases from the Exchange Administrator program.

For more information on circular logging, see your Microsoft Exchange Server documentation.

24.2.2 Exchange Server database restore

A restore obtains backup copies of Exchange databases and transaction logs and returns them to the Exchange Server. To do a restore, the Exchange service and its corresponding database that is being restored must be stopped.

Depending on the backup strategy you choose, restoring an Exchange database might involve restoring multiple backup objects from the Tivoli Storage Manager server.

Upon restarting the Exchange service for the restored database, the Exchange service applies the transactions in the restored transaction logs to the restored database.

24.2.3 Exchange Server database backup delete

This function allows Exchange database backup objects to be deleted from Tivoli Storage Manager storage space when they are no longer needed. This can be done by selecting specific Tivoli Storage Manager stored objects to be deleted, or by specifying to delete all inactive objects older than a stated number of days.

24.2.4 TDP for Exchange security

Standard Tivoli Storage Manager security requirements apply to TDP for Exchange. That is, it must be registered to the Tivoli Storage Manager server and use the appropriate node name and password when connecting to the Tivoli Storage Manager server.

To access the Exchange Server APIs, TDP for Exchange must run under the Exchange Site Services Account. The Site Services Account is the account under which the Exchange services are running and has read/write access to the local registry with backup and restore authority to the Exchange server.

For more information on the Site Services Account, see your *Microsoft Exchange Server* documentation.

24.2.5 TDP for Exchange backup strategy considerations

Depending on your specific requirements regarding network traffic, backup window and acceptable restore times, you might choose to follow different backup strategies. Some commonly used strategies are described below:

- **Full backups only**: This approach is best for Exchange Servers that are relatively small, because it implies that the entire database is backed up each time. Each backup takes longer to perform, but the restore process is most efficient because only the most recent (or other appropriate) full backup needs to be restored.
- **Full backup plus incremental backups**: This strategy is commonly used when the normal backup window or network capacity cannot support a full backup each time. In such cases, a periodic full backup followed by a series of incremental backups allows the backup window and network traffic to be minimized during peak usage times. An example of this is a weekly full backup, done on the weekend, that is followed by daily incremental backups. The full backups can be done during low usage times when a larger backup window and increased network traffic can be tolerated. The restore process becomes more complex, however, because a full backup, as well as subsequent incremental backups, must be restored.

  You should also consider setting the Tivoli Storage Manager policies to ensure that all of the incremental backups are stored together (collocated). This helps improve restore performance. This can reduce the number of media mounts necessary for restoring a series of incremental backups, which should result in faster restores.
- **Full backup plus differentials**: This process provides an easier restore than the full plus incremental backup. This approach might be useful if your backup window and network capacity are sufficient to handle the backup of all transaction logs that accumulate between full backups. The reason for this is because it requires the transfer of only one differential plus the last full backup to accomplish a restore. However, the same amount of data needs to be transferred in the one differential image as in the series of incremental backups.

  Therefore, a full backup plus differential backup policy results in more network traffic and more Tivoli Storage Manager storage usage. This assumes that the differential backups are done with the same frequency as the incremental backups.

  You should carefully consider whether there is sufficient advantage to justify the additional resource necessary to resend all prior transaction logs with each subsequent differential backup.

Part 5. Tivoli Storage Manager and complementary products

Part 5 covers the relationship of Tivoli Storage Manager to some complementary products.

Chapter 25. Complementary products and clients

Figure 109 shows the company logos of many of the complementary Tivoli Storage Manager products and clients.

Figure 109. Complementary Tivoli Storage Manager products and clients

In this chapter we describe some vendor and IBM products that supplement Tivoli Storage Manager. Some products, such as ETI's Tandem BackHome! use the API to access and use Tivoli Storage Manager as a data repository. We also include a service offering for Fujitsu clients, additional database backup products and some other functions which Tivoli Storage Manager may take advantage of, such as remote copy.

25.1 Tivoli Storage Manager clients and terms

What are the differences among Tivoli Storage Manager clients, Tivoli Data Protection application clients, Tivoli Storage Manager service offerings, and Tivoli Storage Manager complementary products?

25.1.1 Tivoli Storage Manager clients

Tivoli Storage Manager backup/archive clients are shipped on CD with the Tivoli Storage Manager server code. They provide the full function of Backup, Restore, Archive, and Retrieve. Tivoli Storage Manager client code is provided free of charge, however, their use must be licensed on the server.

25.1.2 Tivoli Data Protection products

Tivoli Data Protection (TDP) products are separately priced software which use the Tivoli Storage Manager API to connect to the Tivoli Storage Manager Server. These products provide the interfaces to backup and restore specific application software to a Tivoli Storage Manager server. Examples are TDP for Lotus Notes and TDP for Oracle. For more information on supported applications, please see:

`http://www.tivoli.com/products/index/storage_mgr/addbase.html`

25.1.3 Tivoli Storage Manager service offerings

Tivoli Storage Manager service offerings are additional, special-use products which use Tivoli Storage Manager. They are made available under specific conditions and support agreements only. Some examples of service offerings are the UXP/V client mentioned in this chapter and ADINT for backing up SAP Adabas databases (see 20.2, “Methods of backing up R/3 databases” on page 190).

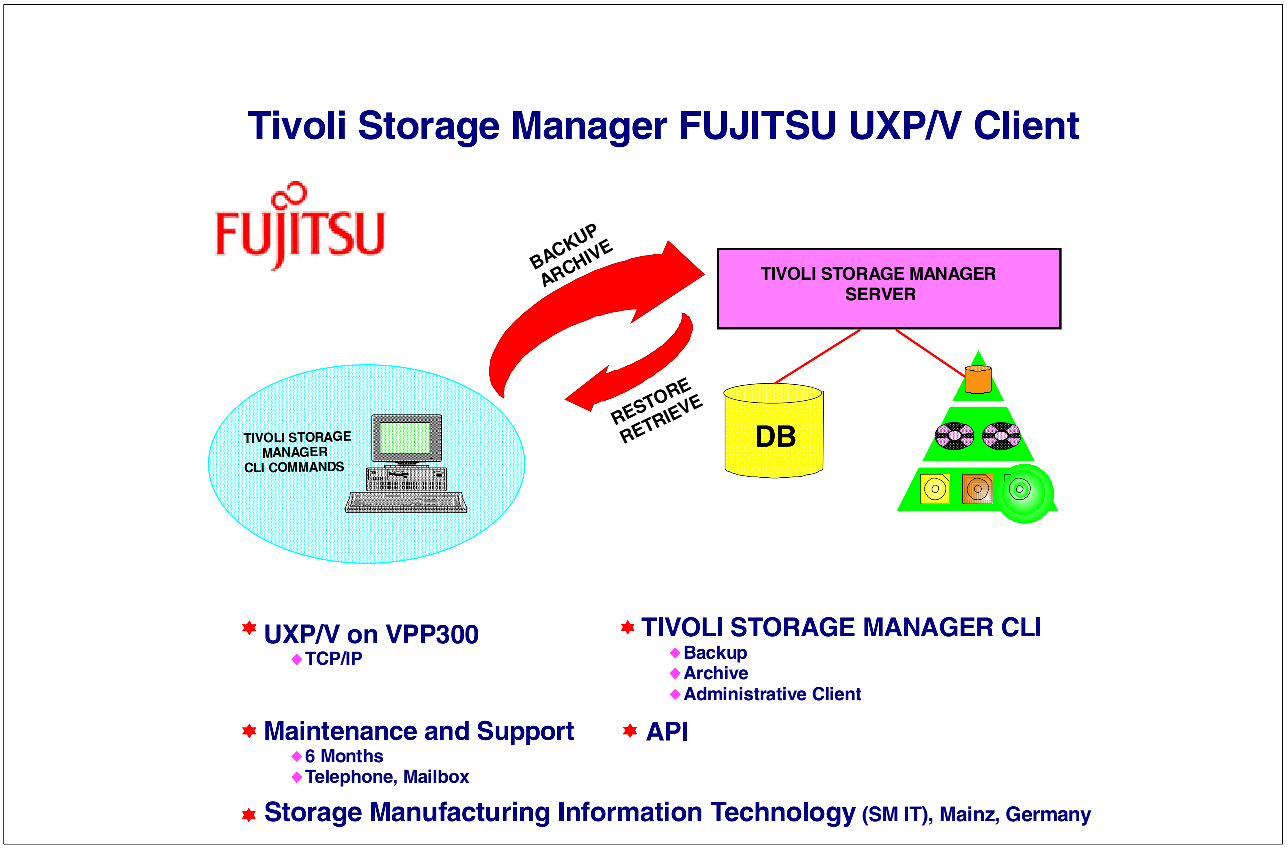
25.1.4 Tivoli Storage Manager Complementary Products

Tivoli Storage Manager complementary products are products that are produced and supported by an IBM Business Partner or IBM Alliance agreement. These products connect to Tivoli Storage Manager using the Tivoli Storage Manager API and usually use their own proprietary backup and recovery software method. They are sold and supported by the owning company.

25.2 Tivoli Storage Manager Fujitsu UXP/V Client

The Tivoli Storage Manager Fujitsu UXP/V client is shown in Figure 110. It is available as a service offering from IBM (SM IT) Mainz in Germany.

The service offering provides both features and maintenance.



Tivoli Storage Manager FUJITSU UXP/V Client

FUJITSU
BACKUP
ARCHIVE
RESTORE
RETRIEVE
TIVOLI STORAGE MANAGER SERVER
DB
TIVOLI STORAGE MANAGER CLI COMMANDS

- UXP/V on VPP300
  - TCP/IP
- TIVOLI STORAGE MANAGER CLI
  - Backup
  - Archive
  - Administrative Client
- Maintenance and Support
  - 6 Months
  - Telephone, Mailbox
- API
- Storage Manufacturing Information Technology (SM IT), Mainz, Germany

Figure 110. IBM Service Offering for FUJITSU UXP/V Client

25.2.1 UXP/V on VPP300

The Fujitsu UXP/V client requires the UXP/V operating system running on a Fujitsu VPP300 series model.

25.2.2 Communication protocol

The Fujitsu UXP/V operating system uses the TCP/IP communication protocol which comes standard with the Fujitsu system.

25.2.3 Tivoli Storage Manager Fujitsu UXP/V Modules

The Tivoli Storage Manager Fujitsu UXP/V client service offering includes the following modules:

- Command Line Module for Backup and Archive Client
- Command Line Module for the Administration Client
- Application Program Interface

25.2.4 Maintenance and support

The service offering also includes the following maintenance and support:

- Maintenance and trouble shooting are provided for the first six months. Maintenance thereafter can be provided at an additional cost.
- A support Telephone line and mailbox are provided for customer problems in IBM (SM IT) Mainz in Germany.

25.2.5 Order Information

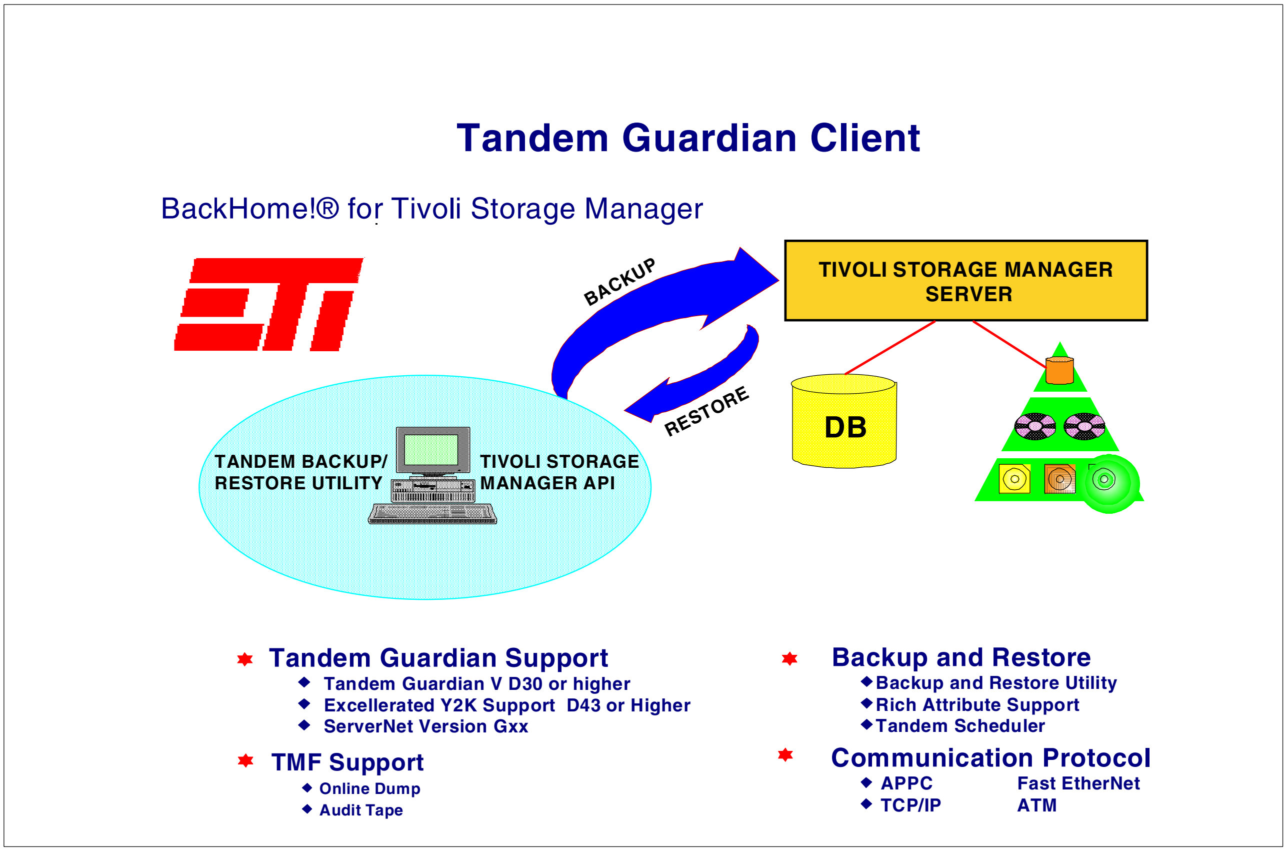
The code is shipped as object code only. For more information, contact IBM (SM IT) Mainz in Germany, or visit their web site at:

`http://www.de.ibm.com`

25.3 BackHome! for Tivoli Storage Manager

Experts en Traitement de l'Information (ETI), Montreal, Canada, offers a Tivoli Storage Manager client solution for the Tandem Guardian platform. ETI's client offering is called BackHome!® for Tivoli Storage Manager. BackHome!® for Tivoli Storage Manager is based on the Tivoli Storage Manager API ported to the Tandem Guardian platform and is incorporated into both ETI's BCOM V4® and BackHome!® products.

Figure 111 shows a summary of BackHome!® and its operation with Tivoli Storage Manager. BackHome!® allows a Tandem system to backup to any Tivoli Storage Manager server and their supported storage hardware.



Tandem Guardian Client

BackHome!® for Tivoli Storage Manager

BACKUP
TIVOLI STORAGE MANAGER SERVER
RESTORE
DB
TANDEM BACKUP/ RESTORE UTILITY
TIVOLI STORAGE MANAGER API

- **Tandem Guardian Support**
  - Tandem Guardian V D30 or higher
  - Excellerated Y2K Support D43 or Higher
  - ServerNet Version Gxx
- **TMF Support**
  - Online Dump
  - Audit Tape
- **Backup and Restore**
  - Backup and Restore Utility
  - Rich Attribute Support
  - Tandem Scheduler
- **Communication Protocol**
  - APPC
  - TCP/IP
  - Fast EtherNet
  - ATM

Figure 111. ETI's BackHome! with Tivoli Storage Manager

25.3.1 What it does

BackHome!® for Tivoli Storage Manager simplifies operations by incorporating Tandem into the Tivoli Storage Manager centralized backup, restore, off site storage and disaster recovery procedures. It:

- Improves utilization of centralized operations staff by reducing requirements for platform specific staffing.
- Extends utilization of high performance tape environments and supports wide range of storage devices.
- Joins the industry's broadest range of client platform support
- Integrates Tandem into centralized Tivoli Storage Manager backup operations
- Operates at channel speed
- Allows multiple concurrent backup and restore operations
- Allows backup or restore to be defined at the volume or file level
- Supports many options for file definition including wildcards, dates and exclusion lists
- Allows backups of TMF Audit Files
- Implemented with minimal operational impact
- Unattended scheduling with conditional follow-on chaining

25.3.2 Specifications

BackHome! can run on any hardware that supports Tandem Guardian (NSK) Version D30 or higher. For Y2K support, Tandem Guardian Version D43 or higher is recommended, along with BackHome! Version 4 or higher.

25.3.3 Communications protocols

ETI supports the following communications protocols:

- TCP/IP
- APPC

ETI supports the following communications cards:

- Servernet
- Token-Ring
- Ethernet
- Fast Ethernet
- ATM

25.3.4 Backup and restore

BackHome!® for Tivoli Storage Manager provides backup and restore of any Tandem file, disk, volume or sub-volume supported by the Tandem Backup/Restore Utility.

BackHome!® for Tivoli Storage Manager uses requests to define a backup or restore function. Each request contains:

- Specification on which files or volumes you want to backup based on attributes delivered to the Backup/Restore utility
- Tivoli Storage Manager server information that tells BackHome!® which Tivoli Storage Manager server to use to store the data
- Specification on which parameter of backup or restore to invoke
- Event logging information

Once the request is complete, it is stored in a request file. The request file can be run immediately or it can be queued or scheduled for later execution.

25.3.5 BackHome!/Tivoli Storage Manager architecture

Figure 112 depicts the architecture of the BackHome!/Tivoli Storage Manager product.

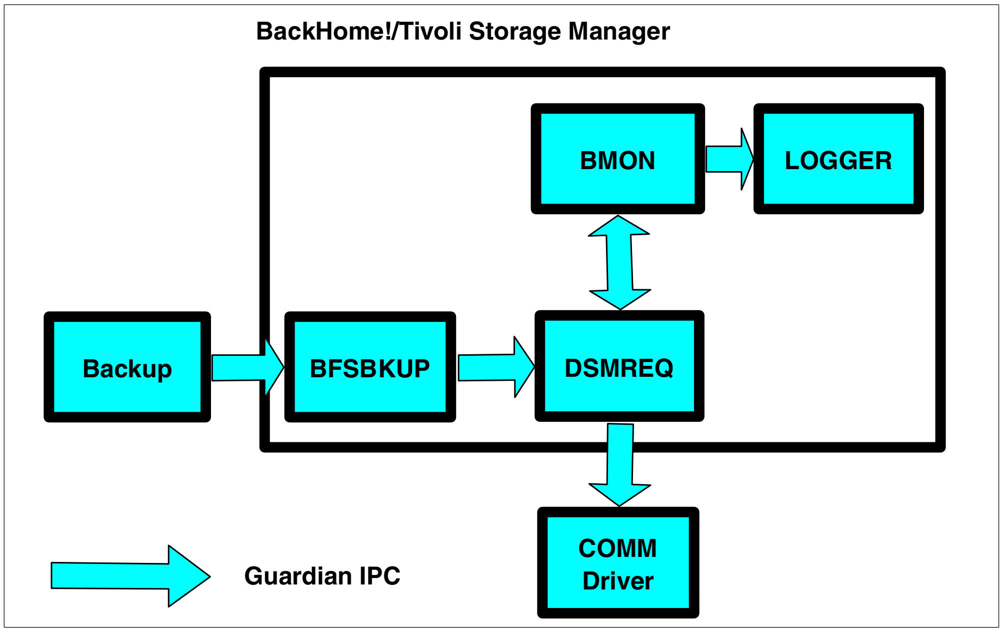


Figure 112. BackHome!/Tivoli Storage Manager architecture

The data is extracted from the disk through the standard Guardian BACKUP utility. Its output is fed to the BFSBKUP process that emulates a tape drive.

BFSBKUP repackages the data and send its output to the Tivoli Storage Manager requestor, which is responsible of sending the data over to TCP/IP via a standard socket. The BCOM Monitor oversees the global operation whereas the Logger handles messages issued by all BackHome!/Tivoli Storage Manager components.

25.3.6 Ordering information

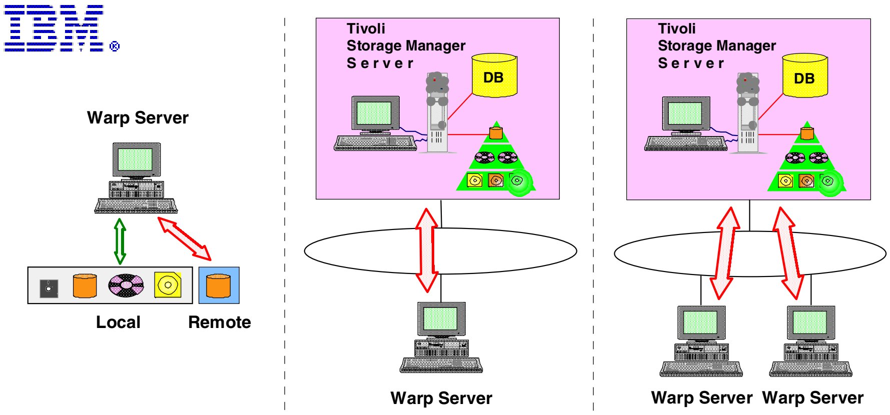
More information, including how to order BackHome!, is available from ETI's Web site `http://www.etinet.com` or by emailing the company at sales@etinet.com.

25.4 OS/2 Warp Server Backup/Restore

OS/2 Warp server data can be distributed across multiple drives and drive types and may have system or security information associated with it. The data may have different backup and recovery requirements. The backup and restore feature that comes with OS/2 Warp Server can be extended by using Tivoli Storage Manager.

OS/2 Warp Server Backup/Restore recognizes Tivoli Storage Manager as a backup and restore facility that provides access to storage devices. OS/2 Warp Server Backup/Restore regards Tivoli Storage Manager as a unique type of storage device that enables access to all Tivoli Storage Manager devices. It treats the Tivoli Storage Manager device as a single, fixed volume that is accessed through a communications link. This is shown in overview in Figure 113.

OS/2 Warp Server Backup/Restore and TSM



- **Warp Server**
  - **Backup/Restore**
- **Warp Server**
  - **Backup/Restore As a Tivoli Storage Manaer API Application**
- **Tivoli Storage Manager Backup Archive Client**

Figure 113. OS/2 Warp Server

25.4.1 Warp Server Backup/Restore

The OS/2 Warp Server Backup/Restore utility manages backups for data on:

- Diskette drives
- 386 HPFS formatted drives
- HPFS formatted drives
- FAT formatted drives
- Long file names
- Extended attributes
- ACPs
- Locally attached hard drives
- LAN Alias hard drives
- Remotely attached hard drives
- Locally attached manual change tape drives
- Locally attached manual change optical drives

The utility provides default backup and restore routines and a GUI for quick and easy setup of backup, restore, and recovery procedures.

25.4.2 Warp Server Backup/Restore as an API application

The OS/2 Warp Server Backup/Restore utility as a Tivoli Storage Manager API application is particularly appropriate in small environments as it combines exploitation of Tivoli Storage Manager with simple setup and management. The backup/restore utility is responsible for the actual backup and recovery of the data, but the data is stored in Tivoli Storage Manager storage pools. As soon as the data is associated with the Tivoli Storage Manager storage media, the

Backup/Restore utility releases control of the data, and the Tivoli Storage Manager server is responsible for managing and protecting the data. Tivoli Storage Manager can move the data between its storage volumes according to the storage hierarchy and migration thresholds specified.

25.4.3 Tivoli Storage Manager backup/archive Client

In an environment with a small number of servers and a small amount of data to manage, the OS/2 Warp Backup/Restore utility is ideal. As the environment grows, and the number of clients and/or amount of data increases, using the standard Tivoli Storage Manager OS/2 client to obtain all of the advantages of a centrally managed Tivoli Storage Manager environment is more appropriate.

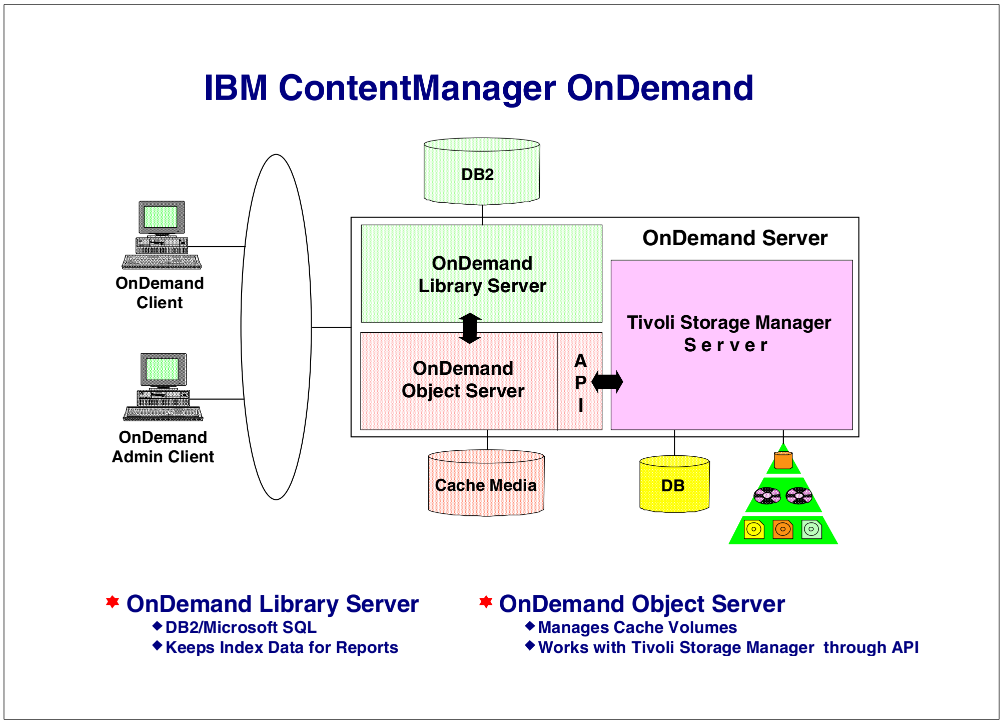
25.4.4 Ordering Information

To order OS/2 Warp Server for Tivoli Storage Manager, contact your IBM representative or IBM Reseller.

25.5 IBM ContentManager OnDemand

IBM's ContentManager OnDemand is a client/server archival and retrieval system for documents that offers an alternative to microfiche and hardcopy storage. It is IBM's strategic offering in the Enterprise Report Management/Computer Output to Laser Disk (ERM/COLD) segment.

OnDemand provides a set of programs to capture, index, store, and retrieve large volumes of computer-generated documents. Once documents are indexed and stored, end users run the OnDemand client GUI under Windows NT/95/98 or OS/2 to search, retrieve, view, print, and fax documents. The OnDemand server is available on AIX, Solaris, HP-UX, AS/400 systems, OS/390, and Windows NT. All of these except the AS/400 and OS/390 versions use Tivoli Storage Manager as the storage management engine, as shown in the overview in Figure 114.



IBM ContentManager OnDemand

- **OnDemand Library Server**
  - DB2/Microsoft SQL
  - Keeps Index Data for Reports
- **OnDemand Object Server**
  - Manages Cache Volumes
  - Works with Tivoli Storage Manager through API

Figure 114. IBM's OnDemand Server

An OnDemand system consists of:

- Client programs
- Library server programs
- Object server programs
- A TCP/IP network

The client programs are the gateway to the report files archived in OnDemand. They allow the end user to construct queries; search for report files; retrieve items from OnDemand; view, print, and fax copies of report files; and attach electronic notes to report file pages.

The OnDemand library server manages a database of information about users and archived report files. The OnDemand object server utilizes Tivoli Storage Manager to manage report files that reside on magnetic and optical storage devices. The OnDemand library and object servers may physically reside on single or separate machines. One library server can have multiple object servers to maintain data storage close to its clients location for rapid retrieval. OnDemand Client, library and object servers communicate over a network running TCP/IP.

25.5.1 OnDemand library server

OnDemand library server uses DB2,or Microsoft SQL (Windows NT) as the database manager to maintain index data for the report files stored in OnDemand. The database manager maintains the OnDemand database with tables that describe the defined applications, application groups, storage sets, folders, groups, users, and printers.

25.5.2 OnDemand object server

OnDemand object server stores its data on either or both ’cache’ and long term storage devices. ’Cache’ refers mainly to hard disks storage and are normally for short term storage as its name implies. It provides for fast retrieval performance to archived data during the initial period. Access to cached data is solely within OnDemand applications. For long term storage, OnDemand object server utilizes Tivoli Storage Manager as the backend to manage and maintain its data in storage volumes within storage pools of optical and tape devices.

OnDemand communicates to Tivoli Storage Manager via its API for loading and retrieval of data to Tivoli Storage Manager.

25.5.3 Order information

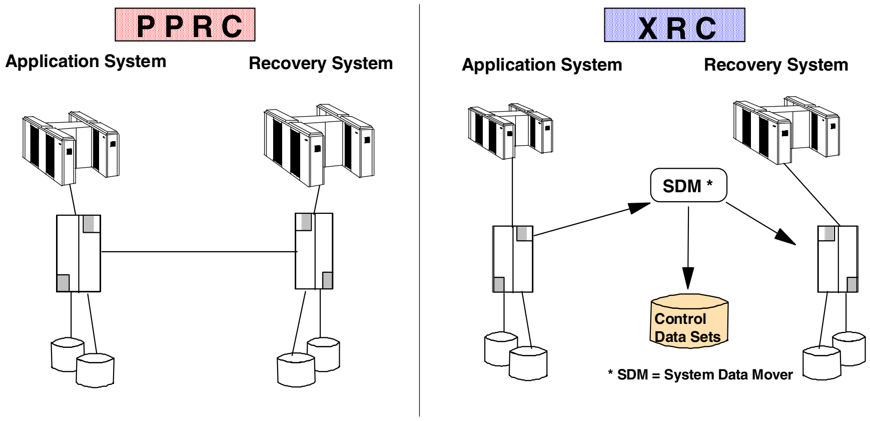
OnDemand can be ordered from IBM. More information is available at the Web site:

`http://www.ibm.com/software/data/ondemand/`

25.6 Remote Copy and Tivoli Storage Manager

Remote copy, shown in overview in Figure 115, is a storage-based disaster recovery and work load migration solution that enables real-time copy of disk data to a remote location. It is an optional function of IBM disk subsystems such as the Enterprise Storage Server (when available).

Remote Copy and Tivoli Storage Manager



- **PPRC**
  - Hardware Solution
  - Synchronous
- **XRC**
  - Hardware and Software Solution
  - Asynchronous
- **Remote Copy and Tivoli Storage Manager**
  - Database and Recovery Log
  - Device Configuration File
  - Volume History File
  - Server Options File

Figure 115. Remote Copy

Remote copy offers two options:

- Peer-to-peer remote copy (PPRC)

- Extended remote copy (XRC)

Both options address the problem of unrecoverable data that occurs between the last, safe backup of an application system to a recovery system and the time when the application system fails.

Disasters occur in many forms. Some occur suddenly and stop all processing at a single point in time; more frequently, others interrupt operations in stages, occurring over several seconds or minutes. The remote copy function addresses the problem caused by intermittent and gradual system failures.

Remote copy is based on two systems, an application system and a recovery system, either in the same building or at remote locations. Each system has specific data that you have identified as remote-copy-managed. Data is written to the primary disk and shadowed to the secondary disk transparently to application users.

25.6.1 PPRC

PPRC is a hardware solution designed for those sites that must have data at the recovery system fully current with the application system and can accept some performance impact on the write operations of applications.

With PPRC, no disk data is lost between the last update at the application system and the recovery at the recovery system.

A PPRC data copy to the recovery system is synchronous with the primary volume's write operations. PPRC does not consider the application system disk write complete until the recovery system write is complete. Each application system write to the recovery system causes an increase in the response time of the application system.

The maximum recommended distance between the application system and the recovery system (for the IBM Enterprise Storage Server) is 103km (60 miles).

25.6.2 XRC

XRC is a combined hardware and software solution designed for those sites that have to maintain the highest levels of performance on their application system and can accept a gap of a few seconds between writes on the application system and writes on the recovery system. There may be little or no data loss if a system failure occurs because the currency of data at the recovery system is only seconds behind the primary write operation.

With XRC, the only data lost is data in transit between the time the application system fails and the recovery system recovers.

XRC allows the primary application's write operation to signal completion before the data is shadowed to the secondary volume.

The maximum distance between the application system and the recovery system is unlimited if channel extenders are used.

25.6.3 Remote Copy and Tivoli Storage Manager

If Tivoli Storage Manager disk data is managed PPRC or XRC, the data is automatically written to the primary as well as the secondary system in real time.

Remote copy is recommended for managing the Tivoli Storage Manager database, recovery log, device configuration file, volume history file, and server options file.

Platform support for the XRC and PPRC functions was originally available on OS/390 platforms only. Additional platforms are planned for support. You should check with your disk vendor for up-to date information.

25.6.4 More information

For more information about remote copy on the IBM ESS, see:

- *IBM Enterprise Storage Server Introduction and Planning Guide,* GC26-7294
- *Implementing ESS Copy Services on System 390,* SG24-5680
- *Implementing ESS Copy Services on UNIX and Windows NT,* SG24-5757

25.7 BMC SQL-BackTrack

BMC Software offers specialized database backup applications for Informix, Oracle, Sybase, SQL Server and R/3 via their SQL-BackTrack set of products. These are available on various operating system platforms to provide a wide support. Each SQL-BackTrack product provides fast physical and raw partition backups, guided recovery, true block level incremental backup, dry-run backup/recovery verification, parallel backup streams, data compression, and encryption. By themselves, the SQL-BackTrack modules direct their backups to a local tape device. When used with the separate OBSI module, the backups are transparently directed to and integrated with, the storage management software product of choice. BMC provide an OBSI module for Tivoli Storage Manager which integrates the database-aware backup and recovery features of SQL-BackTrack with Tivoli Storage Manager. This means that all the objects backed up and restored are managed by the Tivoli Storage Manager server. See Figure 116 for an overview of how it works.

25.7.1 Overview

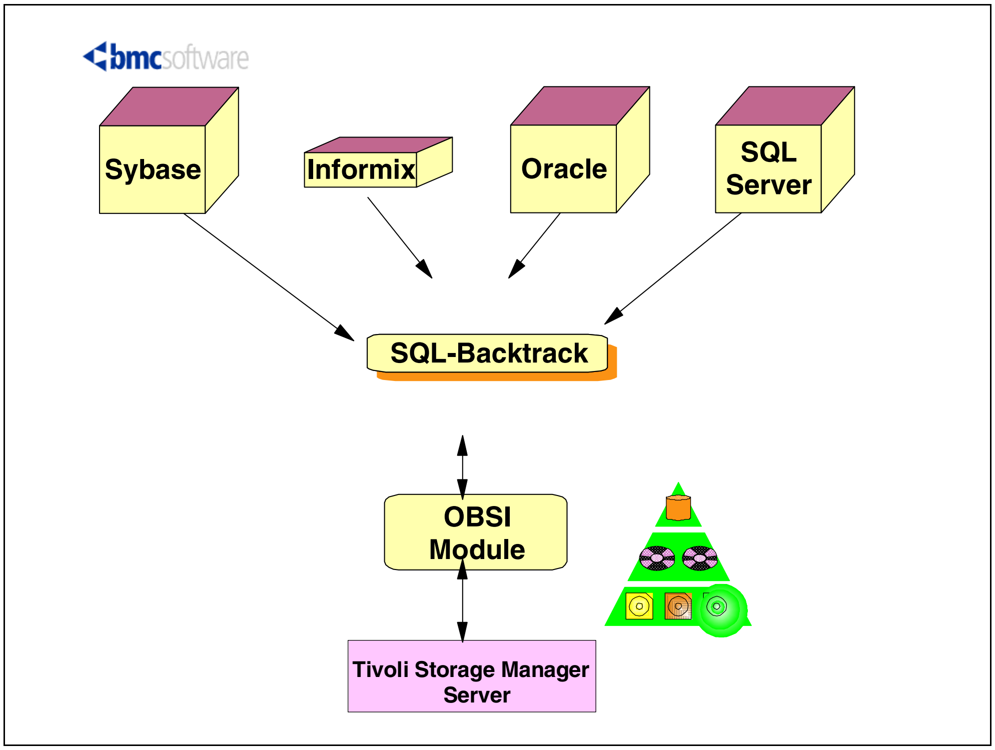


Figure 116. SQL-Backtrack Overview

SQL-BackTrack automates backups, simplifies recovery and provides powerful tools for specialized database administration tasks. It improves backup performance so more data can be backed up in less time, and it greatly simplifies the restore and recovery processes so that database outages cause minimal downtime.

25.7.2 SQL-BackTrack key features

- Fast database backup
- Automated backup and media tracking
- Unattended recovery
- Simplified recovery
- Parallel backup streams/dump striping
- True incremental backup
- Dry-run backup and recovery verification
- Data compression and encryption
- Same interface and procedures across the different databases supported
- Wide operating system platform support

25.7.3 SQL-BackTrack OBSI key features

The SQL-BackTrack Tivoli Storage Manager OBSI module provides:

- Transparent integration of database backups with enterprise storage management system.
- Robotic tape device support for unattended database backups

25.7.4 System requirements

One of the following SQL-BackTrack products:

- SQL-BackTrack for Informix
- SQL-BackTrack for Oracle
- SQL-BackTrack for Sybase
- SQL-Backtrack for MS SQL Server

Plus, SQL-Backtrack OBSI Module for Tivoli Storage Manager and a Tivoli Storage Manager server. The Tivoli Storage Manager server can be on any platform.

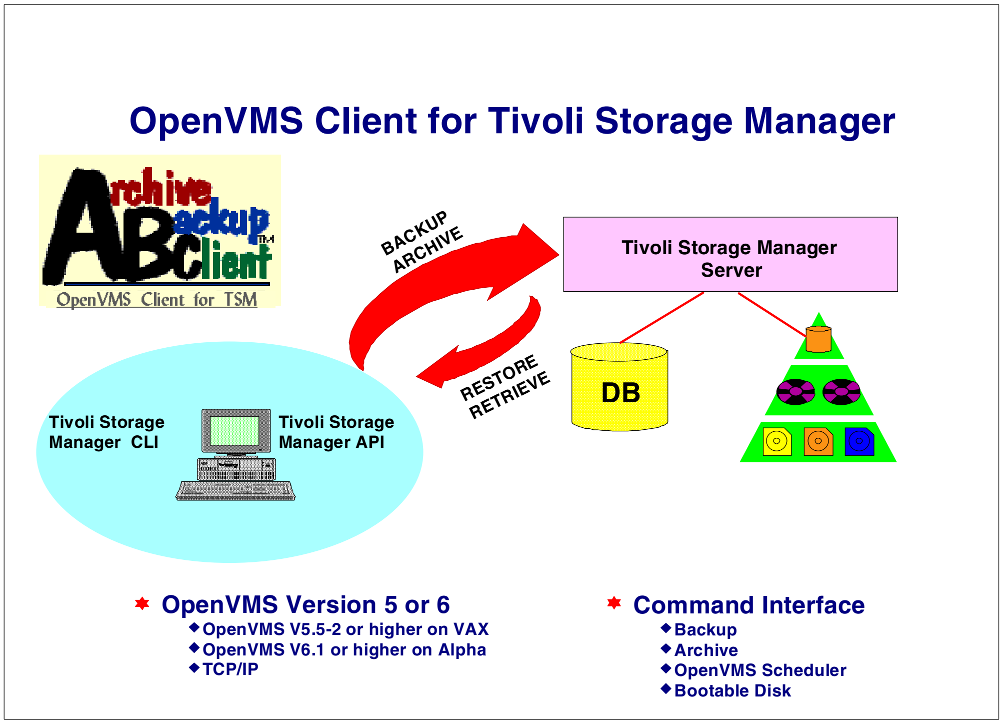
25.7.5 For more information

SQL-BackTrack and the OBSI Module for Tivoli Storage Manager is available both from BMC or their distributors directly, or by reseller agreement through IBM and Tivoli. For more information about SQL-BackTrack, see BMC's Web page `http://www.bmc.com`. It is important to note that if ordering these products, SQL-BackTrack and the OBSI module are separately priced and that both are required to setup backups to Tivoli Storage Manager.

25.8 OpenVMS client

The most recent version of this material is maintained on the ITSO Internet Web pages at:

`http://www.redbooks.ibm.com/solutions/systemsmanagement/tsm/c_vmscli.html`



OpenVMS Client for Tivoli Storage Manager

Archive
Backup
Client
TM
OpenVMS Client for TSM
BACKUP
ARCHIVE
RESTORE
RETRIEVE
Tivoli Storage Manager Server
DB
Tivoli Storage Manager CLI
Tivoli Storage Manager API

- OpenVMS Version 5 or 6
  - OpenVMS V5.5-2 or higher on VAX
  - OpenVMS V6.1 or higher on Alpha
  - TCP/IP
- Command Interface
  - Backup
  - Archive
  - OpenVMS Scheduler
  - Bootable Disk

Figure 117. OpenVMS Client

The Storage Solutions Specialist Inc. (SSSI) Archive Backup Client (ABC) enables Digital OpenVMS users to backup, archive, restore, and retrieve their data through the Tivoli Storage Manager API. This backup-archive client is called the OpenVMS client. See Figure 117.

ABC enables users on OpenVMS systems to make use of the network storage and media management services provided by IBM's Tivoli Storage Manager. Users can backup, archive, restore, query and manage OpenVMS files stored on Tivoli Storage Manager servers as a logical extension to the online OpenVMS ODS-2 file system. This solution makes Tivoli Storage Manager one of the few distributed storage products to provide support for an OpenVMS environment.

An important feature of ABC is its ability to correctly backup and restore disks with complex directory structures, including alias files and directories. This is especially important on an OpenVMS system disk, where there are many alias directories. ABC uses SSSI's own AliasWise technology to manage these alias files.

25.8.1 Cluster environment

ABC is fully supported in any valid and licensed OpenVMS cluster configuration. A separate software license must be purchased for EACH node in the cluster which wishes to use ABC. ABC on any node can be used to backup and restore files on any shared disk in an OpenVMS cluster, including MSCP served disks. The ABC Installation Guide, and ABC User's Guide describes OpenVMScluster operation.

25.8.2 Hardware requirements

Hardware requirements include processors supported and disk space requirements.

Processors Supported

VAX:

- Any valid VAX system configuration supported by and running OpenVMS VAX V5.5-2 or higher

Alpha:

- Any valid Alpha system configuration supported by and running OpenVMS Alpha V6.1 or higher

Disk space required

- During installation: 5000 blocks
- Permanent: 2500 blocks

These counts refer to the disk space required on the system disk.The sizes are approximate; actual sizes may vary depending on the user's system environment, configuration, and software options.

25.8.3 Software requirements

Operating System:

- On VAX systems, OpenVMS VAX V5.5-2 or higher
- On Alpha systems, OpenVMS Alpha V6.1 or higher

25.8.4 Communication protocol and network

ABC uses the TCP/IP network stack. The following TCP/IP implementations have been tested and qualified for the Tivoli Storage Manager API with ABC:

- DEC TCP/IP Services for OpenVMS
- Process Software TCPware
- TGV MultiNet
- Wollongong PathWays

Any TCP/IP implementation which implements C sockets as specified in the DEC C Run-Time Library for OpenVMS Systems, V5.0 or later, is acceptable.

25.8.5 Tivoli Storage Manager Server requirements

Any valid licensed Tivoli Storage Manager server which supports the Tivoli Storage Manager Client API V2.1.2 or higher, and which uses TCP/IP as a supported network protocol. Server support is available on AIX, MVS, HP-UX, Sun, Windows NT Server, and other platforms.

In addition, client connect licenses must be purchased for each ABC client which will use an Tivoli Storage Manager server.

25.8.6 Order information

For further information contact SSSI at this address:

Storage Solutions Specialists, Inc.

PO Box 51313

Colorado Springs, CO 809419-1313

1-888-STORSOL (786-7765)800-290-7172

Email: `sssi@storsol.com`

Web URL: `http://www.storsol.com/`

Chapter 26. Tivoli Data Protection for Workgroups

Topics covered here include the overall functions of Tivoli Data Protection for Workgroups and how these functions integrate with other parts of Tivoli Storage Management.

Pointer to "how-to" information:

An ITSO "how-to" guide is available as a companion to this conceptual information. It provides material describing how to implement and operate this feature or function. It is available as a hardcopy redbook, *Using Tivoli Data Protection for Workgroups,* SG24-5490.

The most recent version of the guide is maintained on the ITSO Internet Web pages at:

```
http://www.redbooks.ibm.com/solutions/systemsmanagement/tsm/tdpfw.html
```

26.1 What does Tivoli Data Protection for Workgroups do?

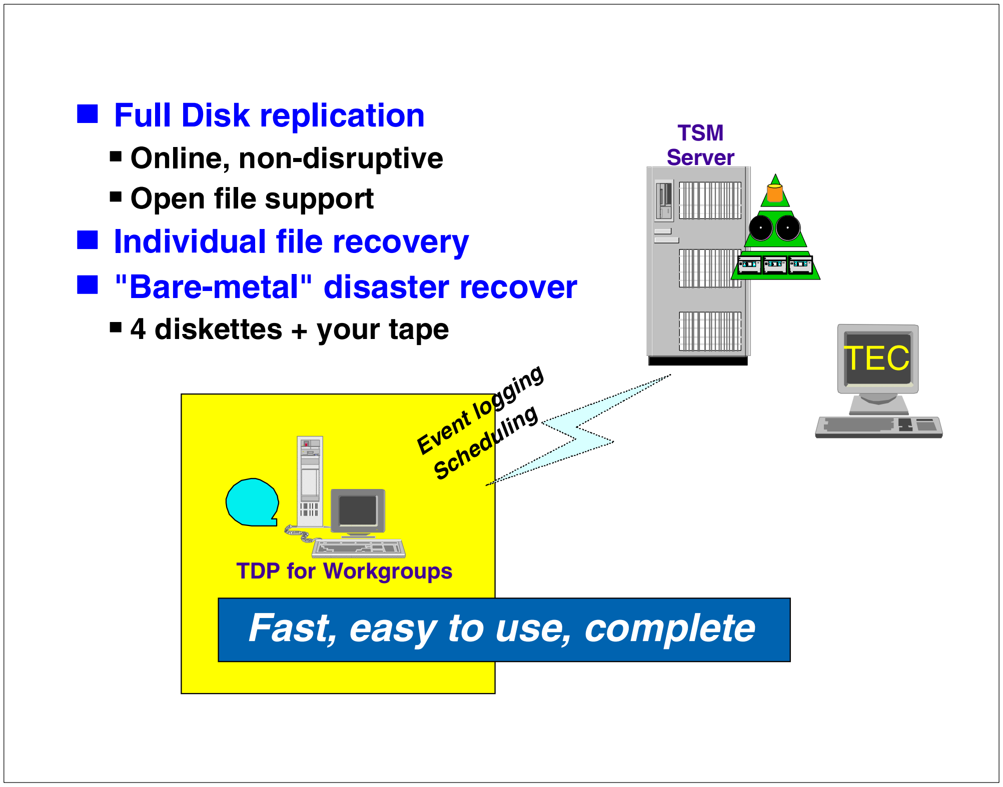
Tivoli Data Protection for Workgroups (TDPfW) provides complete disaster recovery or *bare-metal-restore* for Windows NT machines. It uses a unique Object Replication Technology to ensure backup of all disk contents, such as partition information, boot records, FAT, and NTFS partitions, while allowing full read and write access to the machine during backup. It can back up entire Windows NT machines or volumes at near maximum device speed to a local tape or disk drive. Users do not have to disconnect and applications generally do not need to be shutdown to perform the system replication. Tivoli Data Protection for Workgroups has special caching features to ensure that a true copy of the disk is replicated. For additional products, such as Microsoft Exchange and SQL Server, which have their own internal buffer processing, sample scripts are provided with the TDPfW product, which can be used to ensure a clean backup.

If a disaster occurs, TDPfW can restore the complete machine from the bare metal, including the boot volume, disk partitions, security, operating systems, user files, and even files that were open at the time of the backup, from a locally attached SCSI tape drive. See Figure 118 for an overview. TDPfW is designed for environments where there is little or no IT skill available, therefore, ease of use is a prime feature.

After booting from a set of modified Windows setup disks (four disks in total), it restores the Windows NT system and data as quickly as the tape device can operate. After this, the recovered Windows NT system is ready for rebooting and resuming normal operations. Note that if components of the hardware other than the disks have changed, it is possible that Windows NT may not start due to incompatibilities between the hardware and software configurations. It may be necessary to load additional device drivers. More information is available in the TDPfW manuals.

Tivoli Data Protection for Workgroups is a standalone product which does not prerequisite either the Tivoli Storage Manager client or server. However, if the Tivoli Storage Manager client is also installed and connects to a server, TDPfW can integrate to this environment on two levels, in scheduling and event logging.

First, the replication process can be automated and controlled using the Tivoli Storage Manager client scheduler. The Tivoli Storage Manager client scheduler can start the TDPfW application on a defined date and time for needed functions, such as a replication or single file restore. Second, events from TDPfW can be forwarded to the Tivoli Storage Manager console and from there to any of the Tivoli Storage Manager event receivers, such as the Tivoli Enterprise Console (TEC).



- **Full Disk replication**
  - **Online, non-disruptive**
  - **Open file support**
- **Individual file recovery**
- **"Bare-metal" disaster recover**
  - **4 diskettes + your tape**

Fast, easy to use, complete

Figure 118. Tivoli Data Protection for Workgroups - Overview

Note

Normally, only four disks are needed for disaster recovery. If additional disk drivers were required during the installation of the operating system, an additional diskette is also needed for each disk controller during disaster recovery. The additional drivers added during the Windows NT installation for special disk controllers (for example, RAID, SSA, or Fibre Channel) have to be stored on the additional disk(s). These drivers will be prompted for during disaster recovery (similar to the regular Windows NT installation).

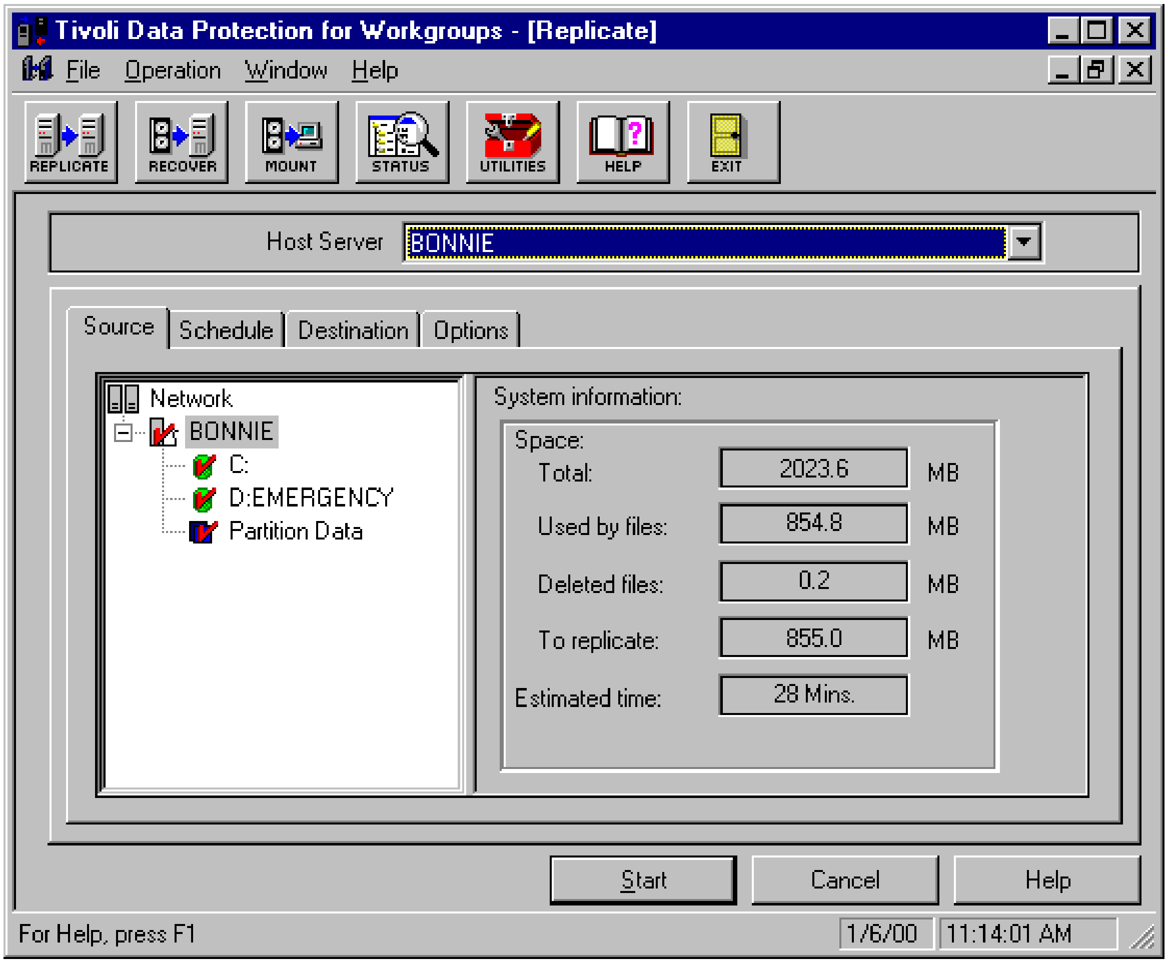
26.1.1 Tivoli Data Protection for Workgroups terms

TDPfW uses some terms differently than their common usage. Examples include *replication*, *TDP Host, mount*, *Partition Data* and *Recover*, which are explained in the following sections.

26.1.1.1 Replicate

Tivoli Data Protection for Workgroups protects an object (server or volume) by doing a logical block-by-block capture of its data to a backup device, usually a tape. The resulting replication is neither an image nor a file-by-file transfer but has the best features of both. Therefore, this is not a traditional backup, and the term *replicate* is used.

TDPfW uses Object Replication Technology to make a logical copy of the source on a destination tape or disk volume. It captures only blocks that contain data and places them unchanged on the destination tape. TDPfW does not have to look at attributes, file names, or file types; it captures or replicates everything. Figure 119 shows a replicate screen.



Host Server: BONNIE

System information:

Space:

Total:	2023.6	MB
Used by files:	854.8	MB
Deleted files:	0.2	MB
To replicate:	855.0	MB
Estimated time:	28 Mins.	

For Help, press F1 | 1/6/00 | 11:14:01 AM

Figure 119. Tivoli Data Protection for Workgroups - Replicate window example

TDPfW can also create a new copy of the volume on the server. If an existing volume is selected, TDPfW overwrites any previous data on the disk. If free space is selected for the copy, TDPfW creates a disk volume with the next available drive letter.

26.1.1.2 Mount

The term *mount* that is used in the TDPfW product and manual, is not comparable to a UNIX mount of a filesystem. This mount is a feature used to see a replicated object (for example a tape volume containing a backup dump of the C drive) as a normal filesystem by using Windows Explorer. Another drive letter will be assigned to this object and normal Windows Explorer functions are available. This means it is very easy to restore a single file or directory to its original or a new location, simply by using the familiar drag-and-drop interface.

26.1.1.3 Partition data

The partition data object encapsulates information collected from the system’s partition tables of all hard disks for later recovery. The partition data object must be included in a replication operation if it is intended to do a bare metal system recovery. This information is not replicated by default therefore it must be explicitly selected, otherwise it will not be possible to perform a disaster recovery from the replication jobs. When the partition data object is selected, TDPfW will pickup the corresponding Windows NT system partition (which is normally the C drive).

26.1.1.4 Recover

This is the normal operation to bring data back to a machine. A partial recovery, where the system is still operational, can be performed either by mounting the required tape and using Windows Explorer, or simply by selecting **Recover** from the TDPfW interface and restoring the required data.

Tivoli Data Protection for Workgroups can recover several types of data.

- File or folder recovery: Individual files, multiple files, and full folders can be recovered where either the full server or the individual volume was replicated to tape.
- Volume recovery: This can be done to the original or to a replacement volume while the system is online. Any replicated volume can be recovered in this way except for a boot or system volume.
- Direct disaster recovery: To recover the entire system (full recovery) or just the boot and system volumes with partition data (minimal recovery), it must be rebooted with the disaster recovery diskettes. From there, the replication tape is read to restore the data.

26.1.2 Growing into a more complete solution

TDPfW offers a very easy-to-use method to protect Windows NT machines against failures of any kind. It is particularly well suited to environments where there are a large number of distributed servers in locations where low or no IT skill is onsite. In this case, TDPfW could be operated in isolation, or, if a central storage management exists, it can leverage that as well. Figure 120 shows how remote systems running TDPfW can integrate into the Tivoli Storage Management environment, where the administrators in the main location can be responsible for scheduling and monitoring the success of the replication operations. In this case, the remote personnel would only need to take care of managing the media used to make the backups and to initiate the restore if and as needed.

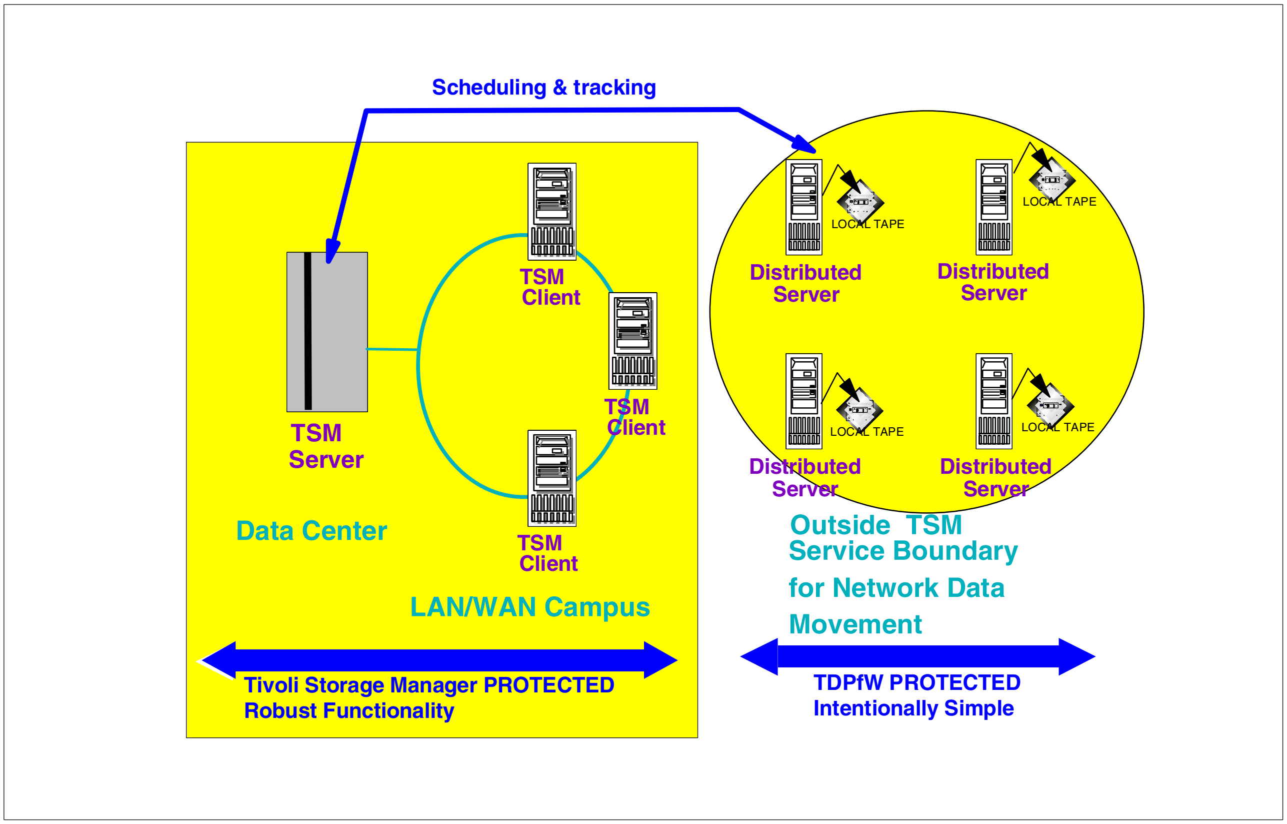


Figure 120. Enterprise integration of TDPfW with Tivoli Storage Management

Chapter 27. Tivoli Storage Manager and AS/400 systems

Pointer to "how-to" information:

An ITSO "how-to" guide is available as a companion to this conceptual information. It provides material describing how to implement and operate this feature or function. It is available as a hardcopy redbook, *A Practical Guide to Implementing Tivoli Storage Manager on AS/400,* SG24-5472.

The most recent version of the guide is maintained on the ITSO Internet Web pages at:

`http://www.redbooks.ibm.com/solutions/systemsmanagement/tsm/c_as400.html`

Tivoli Storage Manager provides a complete data management solution for a heterogeneous network environment, covering backup, archival, space management and disaster recovery for more than 30 different computing platforms and storing this managed data on eight different Tivoli Storage Manager server platforms.

For some years a Tivoli Storage Manager server has been available for AS/400 systems, where it has mainly been used as a repository for end-user workstation and client data. Since the introduction of the BRMS/400 application client for Tivoli Storage Manager, AS/400 data can also be stored and managed using a Tivoli Storage Manager server.

This chapter will discuss on a high level the different aspects of using Tivoli Storage Manager in an AS/400 centric network environment, covering both the application of an AS/400 system as a Tivoli Storage Manager server and the integration of an AS/400 system in an existing Tivoli Storage Manager environment. Also it will give an overview of different solutions when using Tivoli Storage Manager in an AS/400 environment, such as managing data in a partitioned AS/400 system or the application of Tivoli Storage Manager with the AS/400 Integrated Netfinity server.

27.1 Terminology

While BRMS/400 is the strategic data management solution for the AS/400 world, there is also a specific terminology established when talking about managing data in the context. Unfortunately this terminology uses the some of the same terms as Tivoli Storage Manager does, but with different meanings.

Table 12 gives a comparison of terms used in both worlds for the same data management concept. It will be particularly useful for people with AS/400 system and BRMS experience, who want to get familiar with Tivoli Storage Manager or for administrators which have to integrate an AS/400 system in a Tivoli Storage Manager environment.

Table 12. BRMS and Tivoli Storage Manager terminology

Concept	BRMS term	Tivoli Storage Manager term
Saving only changed objects, keep a set number of versions of each object	Incremental Backup with version expiry	Normal incremental backup
Saving all objects, keep a set number of versions of each object	Full backup with version expiry	Selective backup
Saving all objects, keep each object for a set number of days	Full backup with days expiry	Archive
Freeing space by moving files off disk, whilst keeping stub file on disk	Archive, with storage freed	Space Management
Saving objects whilst they are in use	Save while active	Copy serialization

27.2 How does an AS/400 system fit with Tivoli Storage Manager?

An AS/400 system can be easily integrated into a Tivoli Storage Manager data management solution, as either client or server. To act as a Tivoli Storage Manager client, the AS/400 system needs to be running Backup, Recovery and Media Services for AS/400 (BRMS).

27.2.1 What is BRMS?

BRMS is an IBM product for the AS/400 system that uses a set of rules to fully automate the backup and recovery and tape management of an AS/400 system. BRMS will work with a single tape drive, or a multitude of different tape libraries. BRMS can be used in a network of AS/400 systems to share a pool of scratch tapes, although it cannot backup data across a network to another AS/400 system. Figure 121 shows the different elements of BRMS.

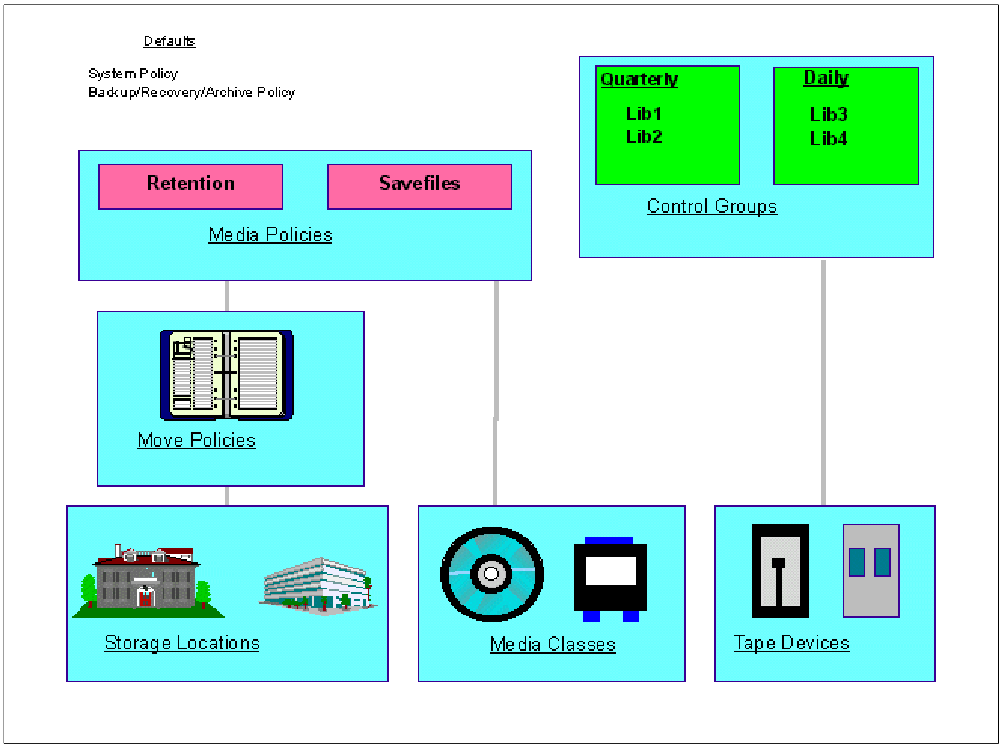


Figure 121. Backup Recovery and Media Services for AS/400

Many AS/400 systems already use BRMS. The Tivoli Storage Manager server for the AS/400 system can be integrated with BRMS, to share tape resources with BRMS if that is required. But the more common application is using the BRMS Application Client to allow the AS/400 system to be a client to a Tivoli Storage Manager server.

With BRMS, a recovery plan can be produced that details exactly what must be done, step-by-step, to recover an AS/400 system after a disaster.

27.2.2 AS/400 system as a Tivoli Storage Manager API client

A set of application program interfaces (APIs) can be obtained for BRMS. These form the BRMS Application Client to a Tivoli Storage Manager server. This allows BRMS to save and restore data on any Tivoli Storage Manager server. See Figure 122.

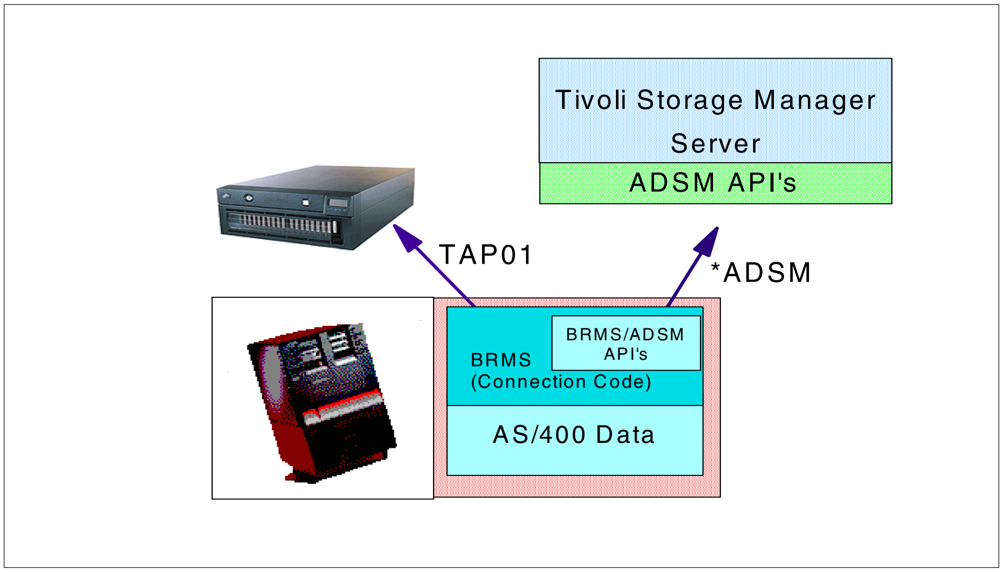


Figure 122. BRMS Application Client

The BRMS Application Client was initially developed to cater for remote AS/400 sites that had little or no AS/400 knowledge. The idea was that once a system administrator had performed a complete backup of the system using BRMS or the native save commands, the user data could then be transferred across a network to a Tivoli Storage Manager server at a remote site. This ensures that the user data is backed up on a regular basis, rather than having to rely on unskilled staff to change tapes and perform backups.

A typical AS/400 system will already be running BRMS to automate the backup and restore of that AS/400's system and user data. With the addition of the BRMS Application Client software, BRMS can be used to save user data to a Tivoli Storage Manager server across the network. System data must still be saved to a local tape device using BRMS or OS/400 commands.

Note it is not recommended to use the Tivoli Storage Manager/400 server and BRMS together to back up a stand-alone AS/400 system. Using Tivoli Storage Manager in this manner is not only more costly, but unnecessarily complicates the backup plan and can degrade backup performance. Best results can be achieved by using BRMS to backup the system directly to local media.

The BRMS Application Client is not intended for use where large amounts of data are to be backed up in short periods of time. If this is the requirement then BRMS should be used to save the data directly to tape.

27.2.2.1 What are the requirements?

OS/400 V4R3 or higher is required to use the BRMS Application Client to save and restore objects from a Tivoli Storage Manager server. Also, the Media Storage Extensions option, and BRMS for AS/400 V4R3 or higher are required. The BRMS Application Client API itself is included on the CD that accompanies the BRMS product, or it can be downloaded from the internet.

27.2.2.2 How does it work?

Backups are defined using BRMS control groups. These are a grouping of objects from the AS/400 system that are to be saved to a specific media and kept for a specific period. A control group can be set up that saves this data to a Tivoli Storage Manager server, rather than to a local tape device. When objects are saved to a Tivoli Storage Manager server they are saved in the form of an AS/400 savefile. A savefile is like an archive that contains one or more other files.

The BRMS interface itself is used to control the backup and restore operations. This is unlike the native Tivoli Storage Manager clients, which all use the same command sets. With the BRMS Application client to Tivoli Storage Manager, backup and recovery commands are issued using either the BRMS menus or native commands. Tivoli Storage Manager appears in these as another possible storage location, along with any local tape devices. Virtually anything can be restored through this interface, from a single AS/400 object to all of the user data contained in many libraries and IFS directories.

Unlike data saved from a conventional Tivoli Storage Manager backup/archive client, the Tivoli Storage Manager server does not know exactly what has been saved to it from the AS/400 client, since the inventory of saved data is kept within BRMS on the client system. All that the server sees is one or more save files. Because of this it is important to set up a special Tivoli Storage Manager policy domain to cope with the specific needs of AS/400 clients, and to ensure that the best use is made of Tivoli Storage Manager server storage space.

27.2.3 AS/400 system as a Tivoli Storage Manager server

Tivoli Storage Manager for AS/400 integrates unattended network backup and archive with disaster recovery planning functions in a single software solution. Tivoli Storage Manager for AS/400 backs up and archives data from more than 30 multi-vendor platforms. In addition to the multiple client platforms, Tivoli Storage Manager for AS/400 supports leading devices and communication protocols. This broad range of support makes an AS/400 server a comprehensive storage management solution.

A typical example of how Tivoli Storage Manager server can fit in with AS/400 systems is the increasingly common distributed computing environments. Here, there may be an application that is divided into three parts. The data for the application is kept on an AS/400 system, the logic for the application may be kept on a Windows NT server, and the user interface for the application is stored locally on desktop PC’s. If a disaster occurs on the logic or data machines then the whole application is unavailable. If a disaster occurs on a desktop PC then just that PC is affected. Tivoli Storage Manager can be used to backup all the desktop PC’s and the Windows NT server to the AS/400 system, creating a complete storage management solution.

27.2.3.1 Why choose an AS/400 system?

The AS/400 system is a very secure, proven and reliable machine, with at least 99.9 percent uptime (according to reported IBM figures) provided by a single AS/400 system with no special high-availability add-on products. This compares very favorably with other operating systems which need to rely on additional high-availability products which may not have the same length of established time in the marketplace.

This high reliability makes an AS/400 system the ideal platform for running mission critical systems, including Tivoli Storage Manager.

27.2.3.2 How does it work?

Tivoli Storage Manager for AS/400 works in the same basic way as any other Tivoli Storage Manager server.

Tivoli Storage Manager for AS/400 supports all OS/400 supported disk drives, a wide range of OS/400 supported tape devices and library, both manual and automatic and optical jukeboxes. The Tivoli Storage Manager for AS/400 support for automated libraries is slightly different to other platforms in that drives are not defined in Tivoli Storage Manager. Drive parameters are sorted out by OS/400 instead.

It is also possible to utilize an external tape management system, such as BRMS, to handle tape libraries as well. When these libraries are defined to Tivoli Storage Manager they are given a media library device parameter, which is used to point to the BRMS device name used for the tape drive or library.

Exit programs are created (the samples supplied work with BRMS without any modifications) that take Tivoli Storage Manager commands and pass them onto the tape management system (such as BRMS) for processing. These commands allow BRMS to handle tape mount, dismount, expiration and deletion processes. This allows BRMS and Tivoli Storage Manager to share a pool of scratch tapes if required.

27.3 Disaster recovery

Tivoli Disaster Recovery Manager is available as an add-on product to Tivoli Storage Manager for AS/400, to aid in the recovery of both Tivoli Storage Manager clients and servers.

The AS/400 client does not support bare metal restore. Recovery of an AS/400 system relies upon the recovery plan available through BRMS. This uses a step by step approach that explains exactly what must be done to recover the entire machine. It automatically uses Tivoli Storage Manager to recover objects wherever necessary.

Recovery of a Tivoli Storage Manager server on an AS/400 system depends on how the AS/400 system has been backed up. Recovering a server running on an AS/400 system is a relatively straight forward process. BRMS (if installed) can be used to aid in the recovery of the server so as to make it a simple step by step procedure.

27.4 Tivoli Space Manager on an AS/400 system

Tivoli Space Manager (HSM) is available as an add-on product to Tivoli Storage Manager for AS/400. The support is the same as for the other HSM platforms, that is, currently it is for Solaris and AIX clients only.

There is no AS/400 HSM Tivoli Storage Manager client. However, this capability does exist in BRMS as dynamic archival and retrieval. This allows certain types of AS/400 objects, such as files, to be archived to storage locations whilst leaving the object description on disk. The object is saved with storage freed as shown in Figure 123. The storage location can be any BRMS defined location, such as a BRMS tape or Tivoli Storage Manager server. When the file is accessed BRMS is invoked to retrieve the data part of the file (the storage freed part), and if the object was last saved to a Tivoli Storage Manager server, then it will be retrieved using the BRMS Application Client.

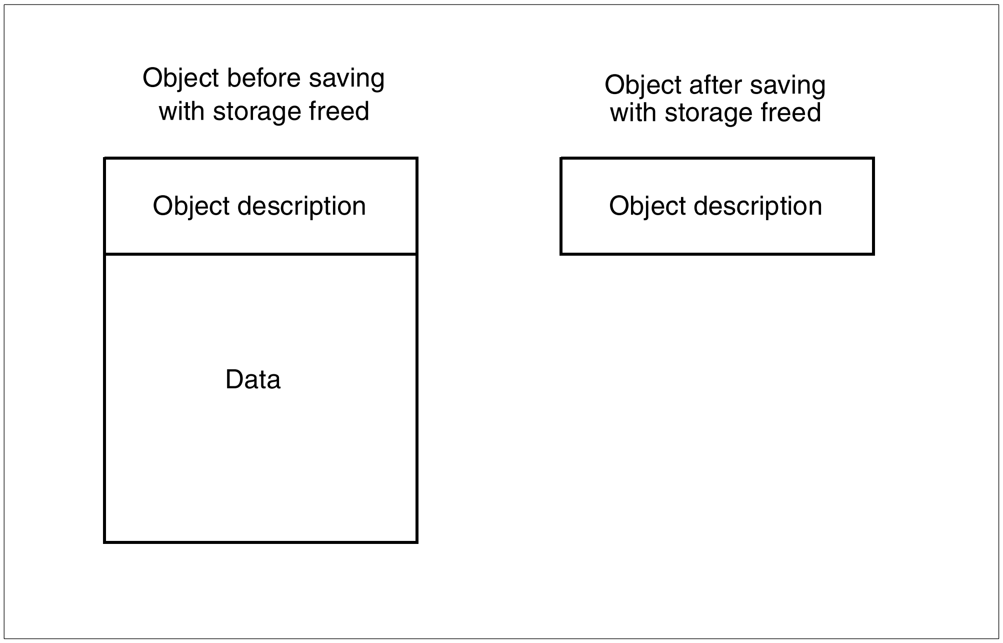


Figure 123. AS/400 object before and after save with storage freed

27.5 Large database and application support on an AS/400 system

The backing up of very large databases can take a long time. That is why IBM, and other vendors, have created various agents that integrate with different database applications. These allow individual records to be backed up instead of the whole database file, saving a lot of time and storage pool space. Online backup is also supported, to reduce or eliminate application downtime.

At the time of writing, all application agents (Tivoli Data Protection for applications) are supported by all Tivoli Storage Manager servers, including AS/400 systems. However, there are currently no agents that run as a client on the AS/400 system.

The BRMS Application Client has not been designed with large database support in mind. If a single byte of information has changed in a database file, then the whole database file has to be backed up.

The recommended method for backing up database files on the AS/400 system is to ensure that they are not in use. With applications such as Lotus Domino, it is preferable that the Domino server is temporarily shutdown whilst the backup occurs, to ensure there are no mail file locks in place and allow correct backup.

Using a function of BRMS it is possible to write into the backup control group that the Domino server should be shutdown before invoking the backup. After the backup has completed another function can be called that starts the server up again.

27.6 Logical partitions

OS/400 V4R4 allows the 7xx, 6xx and Sxx machines to support logical partitions (LPARS). With LPARS is possible to divide up a single physical AS/400 system into smaller logical partitions (LPARS).

The Figure 124 shows a single AS/400 system on the left (SYS A), and an example LPAR configuration on the right (SYS A1, SYS A2, SYS A3). Each LPAR contains a proportion of the total available resource that are available to the single partition AS/400 system on the left (SYS A).

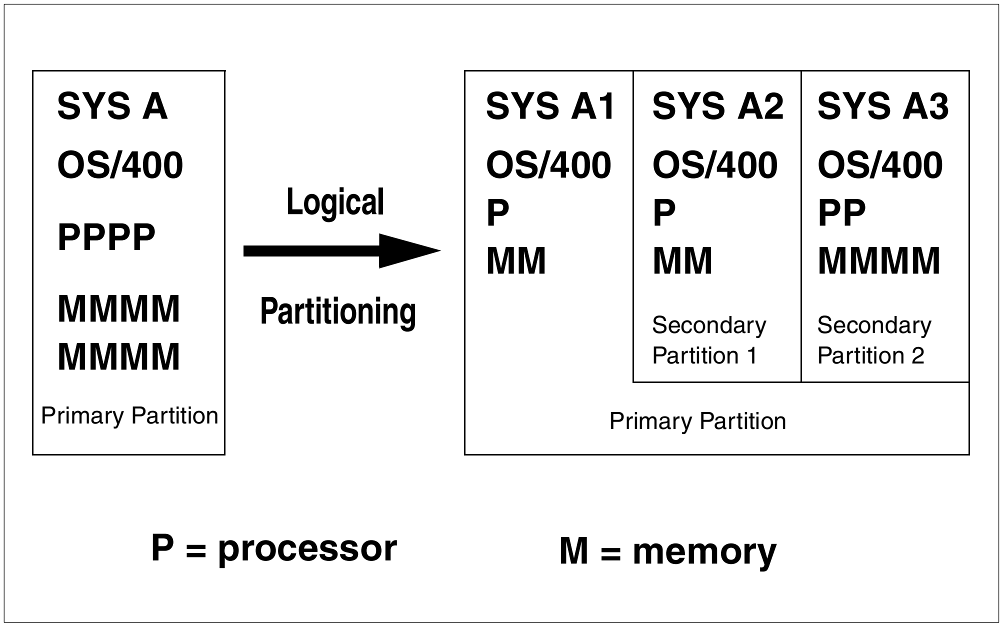


Figure 124. Logical partition overview

Using software called Virtual OptiConnect/400, it is possible to achieve very high data transfer rates between the different LPAR's. With a Tivoli Storage Manager server on one partition and BRMS Application client on another rates of 26GB/hour have been achieved.

This would lend itself very well to installing a Tivoli Storage Manager server on one partition, and backing up all of the user data from the other partitions to the Tivoli Storage manager server at high speed. For more details on Logical Partitioning on AS/400 see the redbook *Slicing the AS/400 with Logical Partitioning*: *a How to Guide*, SG24-5439.

27.7 Integrated Netfinity Server

Many AS/400 systems have an Integrated PC Server (IPCS), or the newer Integrated Netfinity Server installed. These hardware products (shown in Figure 125) allow a Microsoft Windows NT partition to run in the same footprint as the AS/400 system, using different Intel Pentium processors depending on the model.

There are integration features that allow data and resources (such as tape and optical devices) sharing between Windows NT and an AS/400 system.

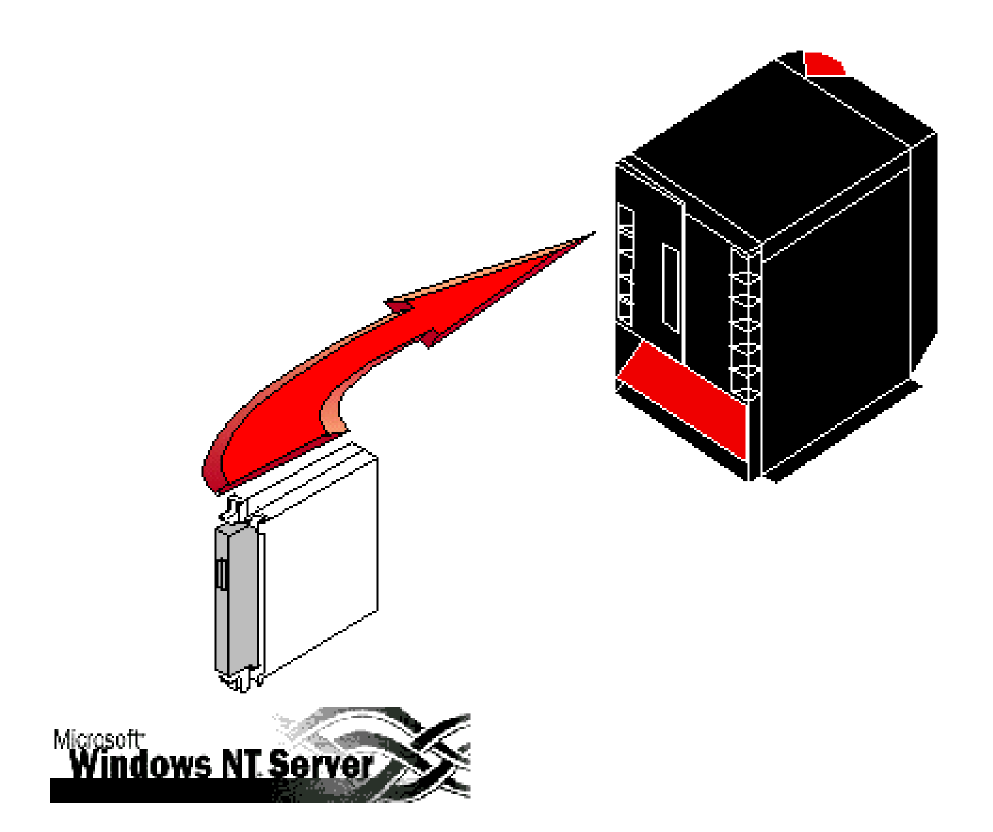


Figure 125. Windows NT server running on IPCS

Careful thought has to be put into the backup and recovery strategy employed when protecting these Windows NT installations. Without Tivoli Storage Manager BRMS (or native OS/400 save and restore commands) can be used to backup the whole server storage spaces at once. However individual objects within the storage areas cannot be backed up and restored without using the built in Windows NT commands. Also, the Windows NT server must be shutdown to backup the server storage areas. This obviously means that users will not be able to access the Windows NT server during the backup operation. It is also feasible that on some smaller Windows NT installations, the network card is shared between the AS/400 and Windows NT. When the Windows NT server is shutdown it also means that the network card becomes unavailable for use by the AS/400 system — rendering LAN access to the AS/400 system impossible.

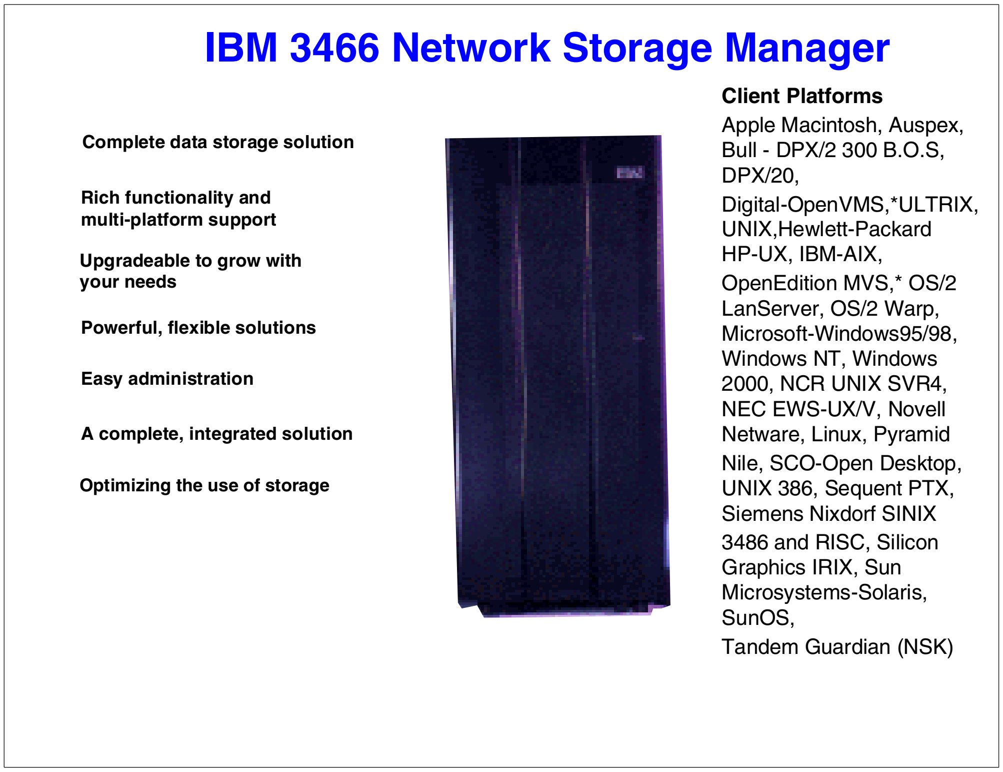
The Tivoli Storage Manager backup/archive client code can be installed on the Windows NT server, just like on a conventional Windows NT machine. This allows individual files and programs to be backed up. The target Tivoli Storage Manager server can be the AS/400 system that has the IPCS installed, or it can be any Tivoli Storage Manager server on the network. It is also possible to run a Tivoli Storage Manager server for Windows NT on the IPCS card.

Chapter 28. Network Storage Manager

The most recent version of this material is maintained on the ITSO Internet Web pages at:

http://www.redbooks.ibm.com/solutions/systemsmanagement/tsm/c_nsm.html

The IBM 3466 Network Storage Manager, shown in Figure 126, is one of the most sophisticated storage management solutions available for the enterprise network. This broad family integrates all of the hardware, software, and storage needed to help protect network distributed data — making it a significant contributor to the long-term viability and success of a business.



IBM 3466 Network Storage Manager

- Complete data storage solution
- Rich functionality and multi-platform support
- Upgradeable to grow with your needs
- Powerful, flexible solutions
- Easy administration
- A complete, integrated solution
- Optimizing the use of storage

Client Platforms

Apple Macintosh, Auspex, Bull - DPX/2 300 B.O.S, DPX/20, Digital-OpenVMS,*ULTRIX, UNIX,Hewlett-Packard HP-UX, IBM-AIX, OpenEdition MVS,* OS/2 LanServer, OS/2 Warp, Microsoft-Windows95/98, Windows NT, Windows 2000, NCR UNIX SVR4, NEC EWS-UX/V, Novell Netware, Linux, Pyramid Nile, SCO-Open Desktop, UNIX 386, Sequent PTX, Siemens Nixdorf SINIX 3486 and RISC, Silicon Graphics IRIX, Sun Microsystems-Solaris, SunOS, Tandem Guardian (NSK)

Figure 126. IBM 3466 Network Storage Manager

28.1 Customer scenario

Consider this real-life scenario: several years ago, a large airline's reservation system went down for 17 hours. This loss of function cost the company five percent of its market share for six months — an amount equivalent to the purchase price of five new jumbo jets. Now consider that 80 percent of all data loss on a daily basis is due to user error — this explains why protecting enterprise network data is so critical. (Source: Information Week Magazine, November 21, 1994.)

28.1.1 The #1 problem

Consider the dilemma facing a large equipment manufacturer. Its operations are run on a corporate mainframe and backed up on a daily basis to tape. Design and engineering systems are handled via a Hewlett-Packard and Sun-based client/server network, with backup routinely conducted by division personnel. Production, distribution, sales, marketing, administration and various other disciplines — which are located both at corporate headquarters and field locations — rely on networked Windows PCs and Apple PowerBook notebook computers, with backup managed on a haphazard, individual basis. There is no centralized strategy for routinely and consistently backing up all of this business-critical information. If an IS audit is initiated, this company wouldn’t have the requisite controls in place to deliver the right information to the audit team. More alarming, there is no disaster recovery plan in place should a fire, flood or other natural disaster occur.

28.1.2 Enterprise-wide data management solution

The IBM 3466 Network Storage Manager plugs easily into this manufacturer’s network — seamlessly handling the Hewlett-Packard, Sun, Windows, and Apple clients. Customers don’t need to fear an IS audit, because this solution provides policy-driven backup and automatic backup scheduling. As a result, they can be assured that the proper controls are in place. And by selecting the Disaster Recovery Manager option, this manufacturer can generate a customized disaster recovery plan to handle nearly any eventuality. This all-in-one solution means that they won’t have to spend lots of time pre-testing components and software—instead, they can get up and running fast.

28.1.3 The #2 problem

For this large, nationwide hotel chain, customer service is the cornerstone of its business. It must have its central reservation system up and running 24 hours-a-day, seven days-a-week — customer satisfaction depends on it. This hotel chain has its central reservation system, a fault-tolerant client/server system, linked via a dedicated high-speed network to a high-end Hewlett-Packard system at corporate headquarters, as well as to smaller networked UNIX-based systems located throughout the country. While the reservation system and data warehouse at headquarters are backed up daily, there is no data backup plan for the other geographic sites and no corporate disaster recovery plan.

28.1.4 24-hour availability solution

Instead of helping to protect just part of a business, the IBM 3466 Network Storage Manager offers a business-critical solution that protects an entire enterprise — enabling it to keep up and running continuously. The IBM 3466 Network Storage Manager can schedule backups around the clock, enabling this hotel chain to take advantage of non-peak times for backup. If data is lost, recovery is fast. The Hierarchical Storage Management (HSM) option also allows them to optimize the use of storage resources and ensure that the central reservation system does not run out of disk space. Also, the Disaster Recovery Manager (DRM) option can help prepare, maintain and execute a disaster recovery plan.

These examples illustrate the importance of keeping business data safe and sound, today and in the future, so that the mission critical knowledge is protected — regularly, consistently and automatically. While this may seem obvious, a disproportionate number of businesses simply don’t have a fail-safe data management strategy for their enterprise network — both for the daily crises that inevitably occur and the catastrophic ones that might occur.

Data loss through system failure, user error, thefts, or natural disaster can cripple a company and cause thousands of dollars in lost revenues, reduced workforce efficiency, and missed opportunities for new products or services. Studies have shown that the average firm loses two to three percent of its gross sales within eight days of a computer outage. If the outage lasts longer than 10 days, more than half of these companies will go out of business within five years. (Source: SunExpert Magazine, July 1996.)

While not all situations involving downtime are as dramatic or as easy to measure, ensuring data availability is critical. Today's business environment expects and depends on rapid data and information exchange within and across networks. So it is understandable that there is no technology investment more essential to long-term success than data backup, recovery, and storage management.

28.2 The IBM 3466 Network Storage Manager

Network Storage Manager is based on IBM's Seascape™ Architecture. Seascape integrates leading technology building blocks from IBM including powerful storage servers and advanced software functionality to provide highly reliable, scalable and versatile storage solutions. Designed with these principles, Network Storage Manager packages together the state-of-the-art components needed to manage and help protect the data in your network. By supporting nearly every networking standard in use today and more than 30 platforms, there is an IBM 3466 Network Storage Manager configuration designed to meet most needs. From a small, privately owned firm to a Fortune 500 enterprise, there is a range of capacity and technology capabilities to choose from.

28.2.1 IBM Network Storage Manager details

Most of the valuable corporate data runs across a network and a company's future depends on finding a reliable way to protect it. That's why the company must be absolutely assured of the best storage management solution possible. Backed by IBM know-how and quality, the IBM 3466 Network Storage Manager is the answer. It provides an all-in-one, powerful, highly-integrated solution designed to manage data. The IBM Network Storage Manager, comes with main frame-quality storage management capabilities for the network computing environment.

Powerful, flexible solutions

Building on IBM's industry-leading storage technology, the IBM 3466 Network Storage Manager comes with a number of built-in hardware and software features designed to provide automated data backup and recovery, advanced storage management, archive, and disaster recovery of mission-critical data stored in today's multi-platform computing environments. Packaged with a RISC processor and award-winning Tivoli Storage Manager software, the IBM 3466 Network Storage Manager can be upgraded to grow with evolving business needs.

All models come with high-speed Serial Storage Architecture (SSA) disks, providing disk capacity ranging from 36 to 864 gigabytes, and industry-leading IBM tape technology options including the Magstar™, the Magstar MP, and the Digital Linear Tape Libraries. These reliable, high-performance tape libraries span from 1 to 57 terabytes of compressed data — giving a flexible range of performance and capacity.

A smart, yet simplified storage management solution

The IBM 3466 Network Storage Manager comes pre-installed with Tivoli Storage Manager client/server software, which provides outstanding data integrity and automated storage management. With the Network Storage Manager, network workstations, desktops, databases, and file servers can be automatically backed up on a routine basis, thereby helping to minimize data loss. The IBM 3466 Network Storage Manager is so smart that it only backs up information that has changed — thereby minimizing network traffic and storage requirements. And because storage administrators can manage the distributed data from a Web-based central location, fewer resources are needed. For enterprises deploying multiple Network Storage Managers, a server-to-server communication feature allows migration of storage pool data between the systems. This reduces overall storage costs by enabling key system resources, such as large tape library systems, to be shared. It is also ideal for “electronic vaulting” of Tivoli Storage Manager database and storage pool backups to a secondary Network Storage Manager for near line disaster recovery. Due to the rich functionality of Tivoli Storage Manager, the planning, preparation and execution of a disaster recovery plan has been greatly simplified as well.

Easy set up and administration

This highly integrated storage management server attaches to a network like any other server, providing a flexible, non-disruptive, easy-to-manage solution. Setup for the end-users and configuration of the storage management is straightforward.

A graphical user interface (GUI) enables administrators to manage the full range of storage management functions — from scheduling backups to adding new clients.

Maintaining the IBM 3466 Network Storage Manager is also straightforward and easy — IBM customer engineers monitor the latest software and hardware fixes and can periodically come on-site to install them.

For added convenience, the IBM 3466 Network Storage Manager comes standard with the Web client and Web admin client interfaces. This feature enables the end users and administrators to use their favorite web browser to conduct storage management functions consistently across your entire enterprise.

Using web browsers, users can back up and restore their own files and verify that particular files are backed up, freeing system administrators to perform other important storage management tasks. Via remote access, mobile users and telecommuters have the same convenience as their in-office peers for backing up and restoring data. With the web browser and the IBM 3466 Network Storage Manager, they can actually download the Tivoli Storage Manager client code onto their own workstations — making implementation easy.

Developing a disaster recovery plan

The Disaster Recovery Manager options is available for the Network Storage Manager. This feature can help reduce the time, cost, and complexity associated with developing and maintaining a well-documented, step-by-step disaster recovery plan for the Tivoli Storage Manager server and clients. It provides a repeatable automated process for preparing, maintaining, and executing disaster recovery — helping minimize downtime and reduce errors during recovery.

Optimizing the use of storage

Most customers place heavy demands on their data storage. That's because complex applications, such as digital imaging, data warehousing, and video-on-demand, produce huge volumes of data, causing systems to outgrow disk storage as fast as it is added. To combat this situation, Hierarchical Storage Management (HSM) is available as an option on the IBM 3466 Network Storage Manager for selected client operating system platforms. HSM enables optimization of storage resources by transparently moving infrequently accessed data that may reside on costly, higher performance storage to less expensive disk, tape or optical. This reduces a company's storage hardware and management costs and gives users the illusion of having infinite disk storage while maintaining high data availability.

A complete, integrated solution

The Network Storage Manager solution supports a broad range of platforms and network computing environments. It works with more than 30 server and workstation platforms from Sun, Hewlett-Packard, Digital Equipment Corporation, and IBM to Microsoft, Apple, Silicon Graphics, Novell, and all other platforms supported by the Tivoli Storage Manager software. The Network Storage Manager also provides broad on-line database support — for example, Oracle, DB2, Informix, SAP/R3 — so databases don't need to go down to be backed up. Furthermore, this solution has been designed to "drop into" an enterprise network with attachments available to Ethernet, FDDI, ATM, and Token Ring interfaces. The bottom line is that the Network Storage Manager integrates proven hardware and software technology to provide backup, archive, and distributed data recovery.

Cutting data management costs and increasing productivity

While the price of disk storage may be decreasing, the price of administering that storage is doing just the opposite. With the IBM 3466 Network Storage Manager, overall computing costs can actually decrease. This is made possible by simplifying the installation and ongoing maintenance procedures so they take less time and demand less in resources. Because all of the components have been pre-tested and much of the configuration has been done, the evaluation, testing and integration cycles are also greatly shortened. Having a solution provided and supported by one vendor means there is one point of contact for support if anything goes wrong.

With its broad platform support, the Network Storage Manager can handle a job that previously required multiple platform-specific solutions

The IBM 3466 Network Storage Manager can also help boost user and storage administrator productivity by automating storage management tasks, providing improved disaster recovery planning, and automatically migrating less frequently used data to lower cost storage media.

This gives mainframe-quality storage management capabilities especially designed for the network computing world — so each company can more efficiently manage and protect its corporate data.

This solution from IBM is powerful, simple, smart, automatic, integrated and cost effective. For more information, contact an IBM representative or visit the Web site:

`http://www.storage.ibm.com/nsm`

28.2.2 IBM Network Storage Manager highlights

- Advanced storage management
- Automated, scheduled and reliable backups
- Optional disaster recovery automation
- Optional hierarchical storage management
- Server-to-server communication
- Enterprise-wide administration, monitoring and reporting
- Easy-to-use GUIs to centrally automate backups and storage management tasks
- Multi-Platform and Protocol Support

  Over 30 server and workstation platforms including HP, SUN, IBM, Microsoft, Apple, Novell Connects to Ethernet, Token-Ring, FDDI and ATM
- Suite of application aware tools for business application such as:

  Oracle, DB2, Informix, Sybase, Lotus Notes, Microsoft Exchange, Lotus Domino, SAP/R3
- Scalability
- Disk capacity from 36 GB to 864 GB
- Tape capacity from 1 TB to 57 TB (compressed)
- Ease-of-use

  Easy-to-use GUIs including the Web client interface allow users to back up and recover their own files

Part 6. Systems management

Part 6 covers the relationship of Tivoli Storage Manager to systems management methods.

Chapter 29. Tivoli Enterprise integration

The most recent version of this material is maintained on the ITSO Internet Web pages at:

```
http://www.redbooks.ibm.com/solutions/systemsmanagement/tsm/c_tivoli.html
```

The Tivoli company was founded to ensure that enterprise-wide computing offers companies a strategic advantage. For customers working in enterprise computing environments, effective systems management is the final barrier to success.

Tivoli Management Software breaks down this barrier by providing a unified, standard management approach to computing environments — one that hides the complexity and scale of these environments.

As a result, Tivoli Management Software has become the standard for managing network computing. Tivoli Management Software products are developed in cooperation with leading hardware and systems software vendors. They are used by leading organizations worldwide, and are supported by dozens of third-party software products.

Its mergers with companies like IBM, Unison, and Software Artistry enable Tivoli to extend and fortify its set of products and services, ushering in a new era in open computing. Tivoli now delivers the first true, end-to-end management solution, which includes a single, standard, unified method for managing everything from desktop PCs and laptops to data center mainframes.

Tivoli Systems is an independent business unit of IBM and is headquartered in Austin, Texas.

This chapter presents an overview of Tivoli Enterprise, showing where Tivoli Storage Manager fits into the systems management organization and how it integrates with other Tivoli products.

A detailed coverage of Tivoli Enterprise can be found in the redbook *An Introduction to Tivoli Enterprise*, SG24-5494.

29.1 Tivoli Enterprise overview

An enterprise consists of a large number of resources to manage. These resources can be network components, operating systems, databases, middleware, and off-the-shelf or custom applications.

The foundation of the Tivoli Enterprise architecture (Figure 127) is a distributed object- oriented software called *Tivoli Framework*. Most of the applications of the Tivoli Enterprise suite use the services included in the framework. This means that when a major function in the framework is improved, these Tivoli applications can take advantage of the improvement. The Tivoli Framework also serves as a single point of integration for the Tivoli and third-party applications.

In addition to the framework, Tivoli Enterprise provides a suite of management products in the disciplines of deployment, availability, operations, and security management. We often call these applications the fundamental Tivoli management applications or the Tivoli core applications.

Tivoli Enterprise also provides a number of modules to manage specific mission critical applications such as:

- SAP R/3 and MQSeries
- E-mail systems, such as Domino/Notes and Microsoft Exchange
- Internet servers, such as Microsoft Commercial Internet System servers and Netscape Suitespot servers
- RDBMS (Relational Database Management System), such as Oracle, Sybase, DB/2, Informix, and MSQL

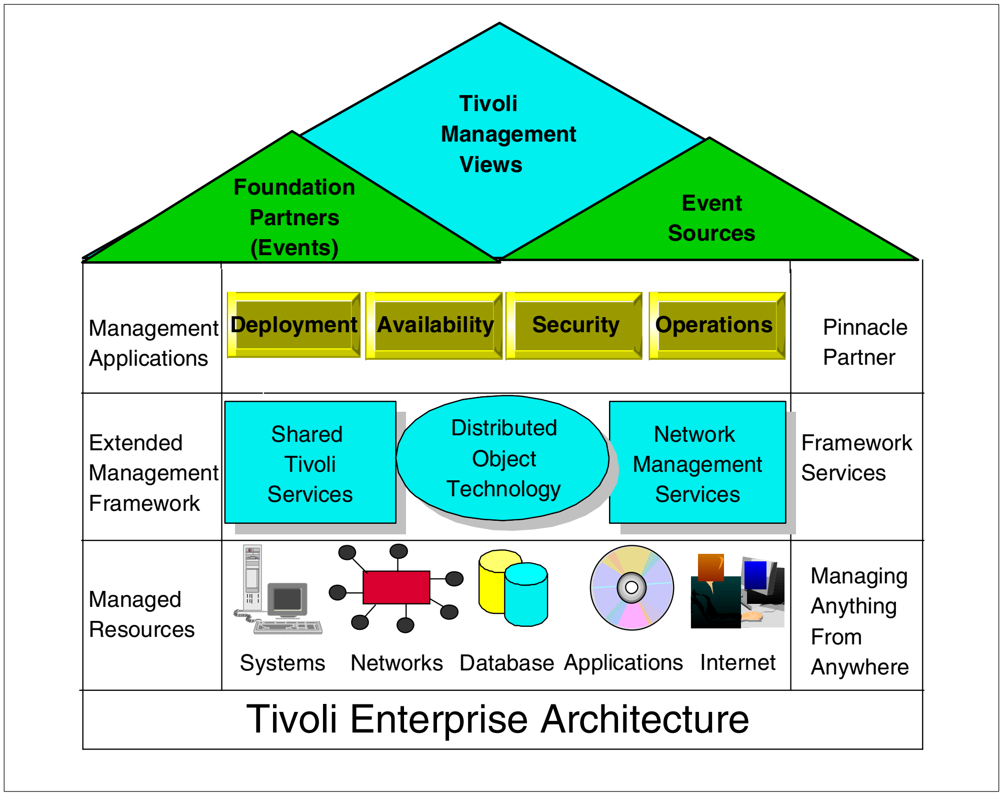


Figure 127. Tivoli Enterprise Architecture

29.1.1 Tivoli Enterprise modules

This section provides an overview of each of the the principal Tivoli Enterprise modules.

29.1.1.1 Tivoli Framework

The Tivoli Framework provides the basic system management services, such as communications, presentation, security, and so on, that most of the Tivoli Management applications use, which ensures consistency and integration. At its core, the Tivoli Framework provides the facilities to transfer files and execute commands on remote systems with built-in security and authorization roles. The Tivoli Management applications can use these core facilities to implement management functions, including software distribution, resource monitoring and system configuration.

Most Tivoli systems management tasks, regardless of the application or component that is to be managed, may be performed by using the Tivoli desktop that provides a user interface consistent throughout management applications.

In the upper left corner of Figure 128 are the main Tivoli Framework products and their logical connections to each other. In the lower part is the Tivoli Storage Manager suite with the server itself, and the different clients and special modules. On the right side are some applications which be managed by Tivoli Management modules and by the Tivoli Data Protection modules for applications.

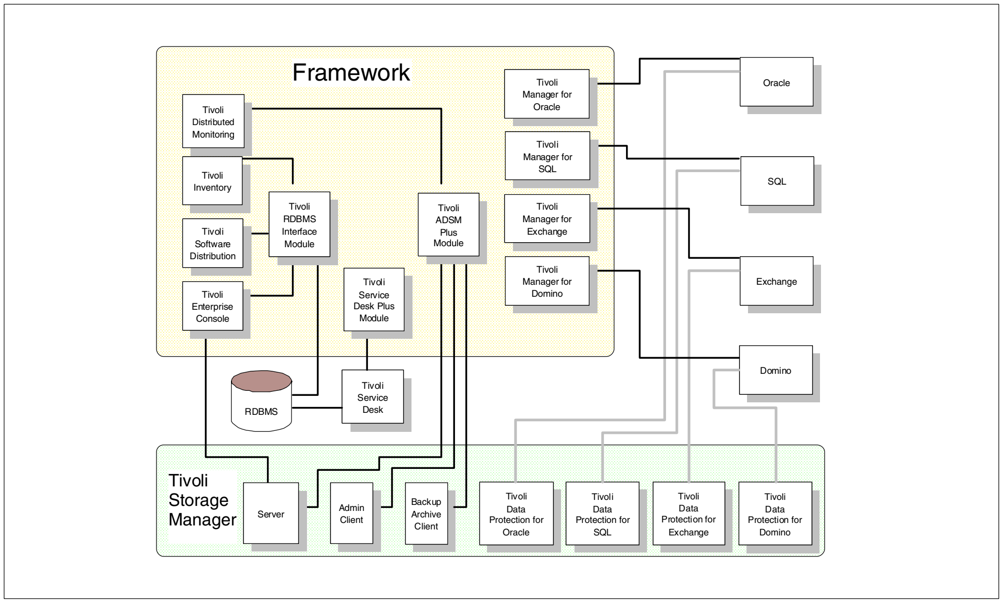

Framework
Tivoli Distributed Monitoring
Tivoli Inventory
Tivoli Software Distribution
Tivoli Enterprise Console
Tivoli RDBMS Interface Module
Tivoli Service Desk Plus Module
Tivoli ADSM Plus Module
Tivoli Manager for Oracle
Tivoli Manager for SQL
Tivoli Manager for Exchange
Tivoli Manager for Domino
Oracle
SQL
Exchange
Domino
RDBMS
Tivoli Service Desk
Tivoli Storage Manager
Server
Admin Client
Backup Archive Client
Tivoli Data Protection for Oracle
Tivoli Data Protection for SQL
Tivoli Data Protection for Exchange
Tivoli Data Protection for Domino

Figure 128. Tivoli Enterprise Framework, applications and Tivoli Storage Manager

29.1.1.2 Tivoli Distributed Monitoring

Tivoli Distributed Monitoring is an application that allows monitoring of the status of a wide range of geographically-dispersed platforms from different vendors running different operating systems including resources that are not part of the Tivoli Environment.

A monitor is an entity that controls specific aspects of a resource (percentage of disk space, status of a print queue, database process status, load average of a system, network collisions, and so on). Its definition contains threshold values and various response actions triggered upon reaching a threshold. Tivoli Distributed Monitoring uses the concept of management by subscription, in common with the other Tivoli core applications. Monitors are defined centrally in distributed monitoring profiles which are then distributed and activated on the target systems.

Tivoli Distributed Monitoring provides the network computing environment with the following features:

- Centralized, synchronous (scheduled) monitoring of remote resources
- Predefined monitors for almost every resource (monitoring collections)
- Strong mechanism to generate events and alarms
- Automated decisions and actions in response to alarms or events
- Various responses (e-mail, triggering a program)

- Custom scripts for monitoring specific applications
- Full integration with the Tivoli Enterprise Console event server
- Data collection for statistical analysis and capacity planning

Tivoli Distributed Monitoring provides a variety of predefined monitors which can be customized for a particular environment. In the Tivoli Storage Manager environment, one or more profiles can be defined. These profiles contain one or more monitors which define the particular object being monitored and the actions to perform when it occurs. Monitors can be defined for example, which will check that the Tivoli Storage Manager server daemon is running or check for the results of a server script or SQL query execution. A monitor could also check the utilization of various resources, such as paging space or free disk space on the server. Once the profiles and its monitors are defined, one or more Tivoli Storage Manager servers will be subscribed to it, so that the monitors will run.

29.1.1.3 Tivoli Software Distribution

Tivoli Software Distribution provides facilities for the distribution and installation of software to managed systems in a Tivoli environment. Tivoli Software Distribution uses the facilities provided by the Tivoli Management Framework to distribute file packages in an efficient manner. Administrators use the profile paradigm used by most other Tivoli applications to define file packages to be distributed. These file packages can include any files (executable programs, data files, and so on) and scripts that will be executed before and after the distribution for a proper installation of the files on the target system.

The actual distribution process can use the Multiplexed Distribution (MDist) facility of the framework to optimize the use of the network. MDist is used to define nodes as repeaters so that they become fan-out points for the distribution. By defining an appropriate repeater hierarchy for the network environment, large file packages will only be moved once across slower links, but will still reach multiple target systems.

Tivoli Software Distribution can be setup to distribute Tivoli Storage Manager software packages, for example the backup/archive client or TDP for applications clients. The distribution can include sending the files to the client, automatically running the installation script and performing post-installation customization.

29.1.1.4 Tivoli Inventory

One of the challenges in a network computing environment is keeping track of the hardware and software installed on each machine. Tivoli Inventory addresses this problem by providing the means to gather hardware and software information related to each system and then storing that information in a relational database. Queries and reports can be run to display the information in this database.

Tivoli Inventory has three major benefits:

- It is based on the Tivoli Management Framework and therefore can be tightly and automatically integrated with other Tivoli applications.
- It stores inventory information in a Relational Database Management System (RDBMS) and, therefore, allows any non-Tivoli applications that can access SQL data to share the inventory information. Moreover, it benefits from the advanced features of an RDBMS system, such as scalability and performance.

- Tivoli Inventory has close links with other applications, such as Tivoli Software Distribution, Tivoli Service Desk, and so on.

Tivoli Inventory is extremely flexible and powerful, however some of the ways it could be used to provide integration with Tivoli Storage Manager include:

- Hardware scan so that the specific installed hardware configuration of each Tivoli Storage Manager server (and clients if required) can be scanned, collected and later queried or reported.
- Software scan so that the specific installed software configuration of each Tivoli Storage Manager server (and clients if required) can be scanned, collected and later queried or reported.
- Configuration files scan so that critical Tivoli Storage Manager configuration files can be collected and saved. This provides the ability to restore such files outside of Tivoli Storage Manager, which is useful for example if an options file has been deleted. In this case, a Tivoli Storage Manager restore could not be performed since it requires that option file to be able to start.

29.1.1.5 Tivoli Enterprise Console

For many companies, the computing enterprise is becoming more heterogeneous in nature. It is supporting a wider variety of operating system platforms, communications methods, and a diverse set of applications and databases. Many computing enterprises are also becoming more distributed both from a client/server and a geographical perspective. It follows, therefore, that the computing enterprise is becoming increasingly more demanding to manage and control, and it is getting more difficult to attain acceptable levels of availability. Availability or the lack of availability of the computing resources may be directly related to the bottom line of a company as well as to its competitiveness within the industry.

The people who create and develop the variety of resources that make up a computing environment give the resources the ability to provide status information through the creation and transmission of alarms, messages, alerts, traps, and so on. These may be created in large quantities and may flow through the network expressing significant or insignificant changes in the status of those resources. It is up to the system support teams and operations staff to sort through the potentially large quantity of messages in order to respond appropriately to a given situation.

To issue these problems, Tivoli Enterprise Console (TEC) provides a centralized point of integration and control for enterprise client/server environments. It allows administrators to monitor information about the environments for which they are responsible.

TEC assists in maintaining high availability of the myriad of networks, systems, applications, and databases found within an enterprise. TEC helps detect potential problems before they cause outages. When problems are detected, TEC may be configured to take action and intervene as defined by the customer.

TEC can prevent administrators from being flooded with unnecessary data that masks the real problems. For instance, it can perform automatic actions or filter out duplicate messages. By maintaining a comprehensive history of reported conditions, TEC allows handling only serious problem that happen in a particular time frame or in the context of other previously-received events.

Both TEC and Tivoli Distributed Monitoring (TDM) are capable of monitoring system resources and activities. Both have the capability to respond to events. TEC is more powerful in that it can understand that events reported from different sources are related. Unlike TDM, TEC maintains an event history. The administrators have more power and flexibility to respond to events with TEC.

The scope of events which TEC can monitor is also broader then with TDM. However, most environments use both TEC and TDM. In general, TDM is best suited for monitoring locally and for synchronous monitoring, while TEC is best for asynchronous monitoring. Complex and persistent or unresolved problems within TDM can be forwarded to TEC for further analysis.

29.1.1.6 Tivoli Decision Support

Tivoli Decision Support is a tool that allows systems administrators and managers to make business decisions based on historical data. Tivoli Decision Support consolidates, transforms and presents enterprise management data in an easy-to-use view, which can reveal hidden patterns and natural relationships among the data. Tivoli Decision Support can also help resolve problems. For example, the Decision Support module for Server Performance Management can show how well servers are performing and also show where more resources can best be allocated to improve service levels (additional servers, more memory, and so forth). More information on Tivoli Decision Support and its Storage Management Analysis guide is given in 30.6, “Tivoli Storage Manager and Tivoli Decision Support” on page 281.

29.1.2 Tivoli Data Protection for Applications

Tivoli Data Protection modules allow data in certain databases or applications to be backed up directly to a Tivoli Storage Manager server by using the appropriate module to interface directly with the Tivoli Storage Manager API. Without Tivoli Data Protection (TDP) modules, when data is backed up from the database, the format of this data may need to be converted into files before backup; also, the database may be need to be first taken offline to get a consistent backup.

By using TDP for Applications in conjunction with the Tivoli Storage Manager API, the underlying physical structure of the database (raw devices or files) is handled by the database or application itself. This means that the database or application controls what objects need to be backed up and how to extract them, so it does not matter whether raw devices or files are used. Also, the type of backup (for example, online, offline, incremental, or table space) is determined and controlled by the application.

TDP applications are available for Oracle, SAP R/3, SQL Server, Informix, Notes, Domino and Exchange and are discussed in more detail, from a Tivoli Storage Manager perspective in the previous chapters.

29.1.3 Other Tivoli modules for managing applications

29.1.3.1 Tivoli Plus module for Tivoli Storage Manager

The Tivoli Plus module for Tivoli Storage Manager provides an integration of TME 10 with Tivoli Storage Manager software. This is now available at the Version 3.7 level, which only works with Tivoli Storage Manager Version 3.7.

This integration allows centralized management of the Tivoli Storage Manager application across a multi-platform network. This module provides the following features for managing the Tivoli Storage Manager application:

- Icons for launching the Tivoli Storage Manager application
- Subscription lists for clients and servers
- Monitors for TME 10 Distributed Monitoring
- Tivoli Storage Manager tasks and jobs

29.1.3.2 Tivoli Manager for MS Exchange

Tivoli Manager for Exchange provides a centralized system management tool for Microsoft Exchange servers on the Tivoli platform. Manager for Exchange facilitates the management of Microsoft Exchange Servers on endpoints and the installation of Microsoft Exchange file packs on clients.

Manager for Exchange can be used with the following optional products:

- **Tivoli Enterprise Console (TEC):** Receives, filters, and forwards selected events to an event server for further processing.
- **TEC event adapter for Windows NT (NT event adapter)**: Converts Microsoft Exchange events to the TEC format.
- **Tivoli Software Distribution:** Used to distribute and deploy software across multi-platform networks.

Manager for Exchange provides the following features to enable you to perform system management functions for Microsoft Exchange:

- Tasks to control and automate Microsoft Exchange administrative activities on endpoints
- A selection of monitors for key performance and availability statistics for Microsoft Exchange Servers
- Configuration files for the NT event adapter to receive, filter, and forward selected events to the TEC
- Automated Microsoft Exchange client distribution
- Data input to Tivoli Decision Support

29.1.3.3 Tivoli Manager for MS SQL Server

Tivoli Manager for Microsoft SQL Server complements existing Microsoft tools for Microsoft SQL Server and adds features to manage a large number of database servers in a distributed environment. This product concentrates on those tasks that can be defined and deployed by generic methods. It reduces the complexity of managing large distributed database environments and frees database administrators (DBA’s) to concentrate on the non-generic issues associated with each of the individual servers.

This product provides the ability to manage and monitor Microsoft SQL Servers by providing extensions to TME 10 Framework, Distributed Monitoring, Enterprise Console, and Global Enterprise Manager (GEM). It includes the MSSQLManagerTasks library and the MSSQLServer and MSSQLDatabase monitoring collections.

These tools provide effective management of distributed Microsoft SQL Server computing resources.

The following tasks can be performed using Tivoli Manager for Microsoft SQL Server:

- Register the Microsoft SQL Server with Tivoli
- Automate repetitive DBA operations across all database servers or databases. Tivoli tasks work on multiple servers or databases in a single action.

The tasks provided can:

- Ensure optimal reading of data by monitoring and correcting fragmentation.
- Ensure query plans are based on the correct statistics.
- Ensure space availability on log and data devices.
- Run standard maintenance commands and check for corruption.
- Check database and server configuration settings.
- Keep up-to-date recovery information.
- Back up databases and log files.
- Use the MSSQLServer and MSSQLDatabase monitoring collections for global database server and database availability and performance monitoring facilities. The monitors can be configured to respond automatically to problems. They can run tasks to perform corrective actions.
- Alert administrators to performance problems or serious errors and failures. Monitors can notify administrators in a number of ways. They can also send events to the Tivoli Enterprise Console (TEC).
- Integrate with the Tivoli Storage Manager backup tool to take advantage of features such as secure backups and backup device management.
- Integrate with Tivoli GEM (Global Enterprise Manager) to allow business-system views of managed servers and databases.

Since Tivoli Manager for SQL Server comes preconfigured with tasks to backup the databases and log files, it can integrate easily with TDP for SQL Server to do this backup.

29.1.3.4 Tivoli Manager for Oracle

Tivoli Manager for Oracle - Framework provides management of the tablespaces, datafiles, rollback segments, and redo logs in an Oracle database, and of the sessions and instances associated with a database. Tivoli Manager for Oracle - Framework enables the use of the other Tivoli Manager for Oracle products to manage an Oracle environment. The module consists of the following products:

- Tivoli Manager for Oracle - Framework
- Tivoli Manager for Oracle - Distributed Monitoring
- Tivoli Manager for Oracle - User Management

29.1.3.5 Tivoli Manager for Lotus Domino

The Tivoli Manager for Domino (Manager for Domino) allows the Tivoli or Domino system administrator to manage Lotus Domino within a Tivoli Management Environment (TME). That is, Domino servers can be managed from the same Tivoli Framework desktop as the Tivoli Management Regions (TMRs).

Manager for Domino Monitor Collections offer a wide range of monitoring options, including monitors for all statistics in the events4.nsf database that return numeric values. Program status monitor check the state of Domino add-in tasks.

Manager for Domino makes use of Tivoli Software Distribution file packages to customize and distribute Notes client software.

Manager for Domino uses TEC rules and classes to filter and manage Domino events. Using Event Server rule base functionality, the Manager for Domino rule sets can be extended or modified to meet particular processing requirements. Domino events can also be assigned to an event group, which can then be viewed on one or more Event Consoles.

29.1.3.6 Tivoli Manager for R/3

A typical R/3 environment includes one or more R/3 systems, with each R/3 system consisting of one database server and one or more application servers. The R/3 application servers run on the UNIX and Windows NT operating systems. The database servers run on the UNIX, Windows NT, and OS/390 operating systems.

The Tivoli management environment consists of a Tivoli Management Region (TMR) server, a Tivoli Event Console (TEC) server, and multiple installations of the Tivoli Manager for R/3 product.

The Manager for R/3 operates at both the machine level and the R/3 system level. Management at the machine level is done via monitors provided with the Tivoli Distributed Monitoring product. These monitors provide status on the machines and daemons.

Management at the R/3 system level is done via monitors, event adapters, and tasks provided with the Manager for R/3 product. The Manager for R/3 represents each R/3 system and server as a Tivoli object. Each object is a special type of Tivoli endpoint that enables the Manager for R/3 to distribute monitors and tasks directly to the endpoint. As with any Tivoli endpoint, a task can be run against one or more application servers across the R/3 systems that the TMR server is managing.

The Manager for R/3 provides two event adapters for R/3 application servers. The event adapters are specific to each application server. Each event adapter must be configured for each application server. R/3 application servers are the primary resources used in managing the R/3 system. The application servers obtain most performance metrics. The Manager for R/3 obtains performance information about the R/3 database server through calls made to one of the R/3 system's application servers. This allows the Manager for R/3 to gather database information for all platforms, including DB2 on OS/390.

29.1.4 Tivoli Service Desk

Tivoli Service Desk is a help desk software solution whose primary goal is to improve the efficiency and effectiveness of a corporate service and support operation. It is a suite of three integrated applications:

- **Tivoli Problem Management**: It is the core application of Tivoli Service Desk. This application addresses help desk services related to user requests for problem resolution

- **Tivoli Change Management:** This application addresses the operational changes associated with delivering IT services.
- **Tivoli Asset Management:** Addresses issues surrounding corporate assets.

Tivoli Service Desk integrates with Tivoli Storage Manager by including a menu item in its Tivoli Framework Integration action group to launch the web administrative client. When this is configured, it brings up a prompt for a machine name or IP address of a Tivoli Storage Manager server. It will then start the browser administrative interface on that server system.

29.2 Tivoli Storage Manager and TEC

This section describes how Tivoli Storage Manager integrates with the Tivoli Enterprise Console (TEC).

29.2.1 Tivoli Enterprise Console

Tivoli Enterprise Console is a management-automation application. It collects information from many sources (Tivoli applications, Tivoli partners, network management platforms, relational database systems, customer applications, and heterogeneous operating systems). TEC's grouping and filtering capabilities reduce the number of events displayed to the operations staff.

TEC event adapters capture and filter messages and then forward them as events to the TEC console. Events in TEC are defined by using the Basic Recorder of Objects in C (BAROC) language.

29.2.2 TEC event adapter

Event adapters are small daemon processes which detect that something worth reporting has happened, format the information in a way understandable to TEC, and send the event to the TEC event server for processing.

An event adapter usually resides on the machine where the resource it is monitoring resides. Event adapters are application programs that:

- Detect events of interest.
- Format the events according to the BAROC definition.
- Send the formatted event instances to the TEC event server.

An event adapter has two alternatives for sending TEC events. A "secure" connection makes use of the Object Request Broker (ORB) technology within the Tivoli framework. This method requires that both the originating system and the TEC are Tivoli-managed nodes. The second method is a direct TCP/IP socket connection. This is called an "unsecure" connection by Tivoli, since the originating system does not have to be a managed node. To be a Tivoli-managed node, a system must be running additional Tivoli software (the *oserv* daemon).

29.2.3 Tivoli Storage Manager event adapter

As described in section 30.4.2, “Client and server event reporting” on page 280, Tivoli Storage Manager client and server events can be logged to a variety of internal and external interfaces, called *event receivers*. The TEC is one of these possible receivers.

Since the release of Version 3, Tivoli Storage Manager has included a TEC event adapter as a standard component. This means that the server can send its messages directly to a TEC console if one is available. The Tivoli Storage Manager event adapter uses the unsecure connection method (socket connection) as it provides greater flexibility, supports all Tivoli Storage Manager server operating system platforms (Version 3 and higher), and removes the requirement that the server must be a Tivoli-managed node. If the Tivoli Storage Manager is at V3.7.3 or higher, then messages originating from the TDP for applications modules can also be forwarded to the TEC server. Separate series of numbers are used to identify the messages coming from these sources.

The event adapter for Tivoli Storage Manager is installed by default during server installation. It includes an event definition file called IBMTSM.BAROC (or IBMADSM.BAROC if the server version is 3.7.2 or earlier) which must be imported into a new or existing rulebase on the Tivoli server. The rulebase must be activated, then an event source and an event group defined. Then, the Tivoli Storage Manager server must be configured so that it knows the address and port used by the Tivoli server. Finally, logging must be set to start and events selected to send to TEC. By default, no events will be sent, therefore the administrator can use filtering and selection commands to determine which events should go to TEC. A detailed description on how to set up TEC event logging is found in the appropriate Tivoli Storage Manager Administrator’s Guide.

Chapter 30. Reporting

This chapter covers basic Tivoli Storage Manager reporting and explains which reports may be useful. It also shows the facilities that can be used for reporting which are included with the base Tivoli Storage Manager product. Finally, Tivoli Decision Support for Storage Management Analysis, which is an application guide for Tivoli Decision Support is discussed.

The most recent version of this material is maintained on the ITSO Internet Web pages at:

`http://www.redbooks.ibm.com/solutions/systemsmanagement/tsm/c_reporting.html`

30.1 Why Tivoli Storage Manager reporting?

As with any application, in a Tivoli Storage Manager environment there are a series of tasks that must be performed regularly. To perform these tasks in a timely and efficient way requires a set of resources, such as space in storage pools or in the Tivoli Storage Manager database, and tape drives and volumes. Scheduled operations must complete in a timely manner and without failures. In large environments the number of operations can be quite high, and managing them effectively can be quite complex.

Reports can be very useful for verifying that the tasks set up for Tivoli Storage Manager to perform are carried out in a timely and efficient way.

30.2 Which Tivoli Storage Manager reports are needed?

Reports and their content are very subjective; they depend on the installation’s requirements and on each administrator’s personal approach to use Tivoli Storage Manager. One of the first things to determine is the kind of operations Tivoli Storage Manager performs in each installation. For example, is Tivoli Storage Manager used only to backup files, or is Tivoli Space Manager used also? Is data being archived and are disaster protection procedures in place? The answers to these and other questions will help build a set of reports which is most appropriate to the specific installation requirements.

Describing the exact commands required to generate these reports is beyond the scope of a Concepts Guide; however, they can all be done using standard Tivoli Storage Manager administrative commands and SQL queries. More information on reporting queries is available in the companion redbook *Getting Started with Tivoli Storage Manager: Implementation Guide*, SG24-5416.

30.2.1 Daily summary report

The daily summary report or overview report is the most basic type of report. From this report it can be seen whether Tivoli Storage Manager is doing all that it is expected to do and give a summary of all failed operations, such as missed backups or server errors. The daily summary report should be as short as possible, to facilitate reading and understanding the information. The following information can be shown in a daily summary report:

- List of all scheduled events that failed

- Files that failed during backup
- Summary information about the amount of data transferred to the Tivoli Storage Manager server
- Space usage trends on the Tivoli Storage Manager server
- Removable media (tape) errors
- Server error messages

All information in the daily summary report relates to the last 24-hour period. A report can also be generated once a week. In this case it should contain a summary of all activities and problems that occurred during the week.

30.2.2 Detail reports

Detail reports help explain a particular aspect of the Tivoli Storage Manager environment. There are many possible detail reports, but they can be classified into the following categories:

- **Client activity and traffic reports:** These reports detail the activities performed on individual clients or groups of Tivoli Storage Manager clients. They can show the amount of data transferred to and from the server, categorized by activity type, such as backup, archive, or HSM. They are helpful in identifying delays in client activity and the reason for the delay.
- **Server background processes:** These reports show the activities performed on the server to manage the data that arrives from individual clients. Examples of these activities are migration, reclamation, and storage pool backup. It is useful to know when these activities run and their duration. This information can be used for estimating the impact of new workloads or server processes on current server activities.
- **Server storage pool space utilization:** These reports show server storage pool space utilization by total storage pool space, or by client node, client filespace, or storage pool. They can be point-in-time or trend reports. These reports are useful in determining the amount of storage media required to accommodate current or new workloads.
- **Server database utilization:** These reports can be used to monitor Tivoli Storage Manager server database and log utilization, to ensure that adequate space is always available. If the database or log fills up, Tivoli Storage Manager will not work properly so this should be avoided.
- **Backup/archive schedules:** These reports refer to operations scheduled on the client nodes through the Tivoli Storage Manager scheduler. The reports show scheduled events that failed, which is the basic exception report to monitor, and all scheduled events that occurred. Events scheduled using utilities, such as cron or a third-party scheduling package, are not logged on the Tivoli Storage Manager server, so another reporting mechanism must be used.
- **Administrative schedules:** Administrative operations can be automated with the administrative scheduling facility. Tivoli Storage Manager reports an administrative event as completed when the scheduled operation is started, not when it has completed. You therefore want to get reports that show whether administrative schedules completed successfully.

- **Server configuration:** These static reports show Tivoli Storage Manager server configuration parameters such as management. These can be used as references when analyzing other reports.

30.3 Where is Tivoli Storage Manager information stored?

Most of the information used to create reports is stored on the Tivoli Storage Manager server, but some of it is stored only on client nodes.

30.3.1 Information on the server

On the Tivoli Storage Manager server there are three main sources of information for creating reports:

- **Database:** This is the most important source. The Tivoli Storage Manager database contains all server and most client definitions. It is the prime source of static information. When database information is requested and given, it is in the form of snapshot or point-in-time data; trends cannot be seen.
- **Activity log:** The Tivoli Storage Manager activity log contains all server messages for the last several days. The number of days to keep this information is predefined and can be customized by using a server command. Messages are logged for client sessions, scheduled operations, automated server processes, and any errors that may occur. The information in the activity log is historical information, because the progress of operations over time can be seen.

  With Tivoli Storage Manager, client messages can be logged as events in the activity log. Commands are available to select all, none or a subset of events to be logged.
- **Accounting log:** Tivoli Storage Manager accounting can be activated with a server command. Once started, records are automatically collected and the log is written to a file called `dsmaccnt.log` in the server installation directory. The `dsmaccnt.log` file contains one record for each client session that terminates. The record consists of text and numeric values separated by commas which is a convenient format for loading into a spreadsheet or other data analysis package. It offers summary information for the activities performed during a session, such as files and bytes backed up, archived, or migrated, session wait times, and total data transferred.

30.3.2 Information on the client node

On the Tivoli Storage Manager client there are two main sources of information:

- **Client error log:** On the client node, Tivoli Storage Manager writes error information to the dsmerror.log file, usually for situations where the client did not succeed in contacting the server. By default it is written to the Tivoli Storage Manager client installation directory; however it can be stored on any drive or directory accessible to the client (for example a shared network drive) by setting a client option
- **Scheduler log:** Also on the client node is the dsmsched.log file, which contains information for all scheduled operations, for example, the name of files that are backed up or archived, failures and errors, and backup summary statistics. This file is called dsmsched.log and by default, it is written to the current directory where the client scheduler is started; however it can be

stored on any drive or directory accessible to the client (for example a shared network drive) by setting a client option.

Tivoli Storage Manager client sessions that are not started with the scheduler do not write output to a file by default. Basically the output will be lost unless the messages are saved in some way, for example, by redirecting standard output and errors to a file.

With Tivoli Storage Manager, error and summary information is propagated from the client to the server activity log. Client error and information messages are identified in the server activity log by the prefix characters ANE. This information is stored for both scheduled and nonscheduled Tivoli Storage Manager client sessions.

30.4 Central error logging

Tivoli Storage Manager events can be centrally logged, monitored, and reported by using industry-standard interfaces. Thus, Tivoli Storage Manager implementations can be integrated with system management applications, and centralized control is facilitated.

30.4.1 Central logging of client events

Certain Tivoli Storage Manager client messages can be logged as events on the server. Client messages can be collected in one central point. The intention of client message logging is to log problems encountered during a Tivoli Storage Manager client operation. Therefore only messages indicating an error condition are logged as events. The only exception to this is client backup statistics, which also can be centrally logged.

All events that are to be logged must be enabled by either message number or severity. Enabled client events that are logged to the Tivoli Storage Manager server are, by default, stored in the activity log and displayed on the server console.

30.4.2 Client and server event reporting

To take advantage of standard systems management interfaces, Tivoli Storage Manager provides the ability to send client and server events to external interfaces. Supported interfaces are Simple Network Management Protocol (SNMP) managers such as NetView for AIX, CA Unicenter, or HP OpenView; Tivoli Enterprise Console (see 29.2.3, “Tivoli Storage Manager event adapter” on page 274 for more information); NetView for MVS; the NT event log; a user-written exit; or direct to a file. Interfaces that receive event data are called *event receivers*. Each event message, client or server, can be enabled for any of the supported receivers. It is possible to enable one message or a group of messages for more than one receiver. As with client event logging, events are enabled for receivers by message number or severity.

The concept of redirecting event messages to external interfaces allows Tivoli Storage Manager implementations to be integrated with systems management reporting tools.

30.4.3 SNMP heartbeat monitoring of server

SNMP is also used to monitor network elements from a central point. It enables the monitored systems to send traps notifying the SNMP manager about events taking place on the local system. In addition to sending traps, a heartbeat monitor can be established to monitor whether managed Tivoli Storage Manager servers are still alive. To enable Tivoli Storage Manager to take advantage of SNMP monitoring, it includes an interface for SNMP. The interface is distributed with the server in the form of an SNMP subagent. It is supported for the server running on AIX, HP-UX, Solaris and Windows NT. Communication between the server and the SNMP manager is established through these connection channels: a) Tivoli Storage Manager server-SNMP subagent, b) SNMP subagent-SNMP agent, and c) SNMP agent-SNMP manager. To enable SNMP subagent-SNMP agent communication, the SNMP agent must support the Distributed Protocol Interface (DPI).

30.5 SQL queries and ODBC interface

Tivoli Storage Manager provides an SQL interface which supports queries to its internal database. The interface is read-only and includes a SELECT command and an open database connectivity (ODBC) driver for Windows NT.

30.5.1 SELECT command

The SQL interface represents Tivoli Storage Manager information in the form of relational tables containing rows and columns that can be accessed by the SELECT command. The SELECT command uses standard SQL syntax compliant with the SQL92/93 standard and can be used only on the administrative command line client.

Because SQL processing uses database resources, long-running or very complicated select statements can slow down server performance significantly. Therefore resource-intensive queries display a confirmation message, offering the possibility to abort the query before executing it.

30.5.2 ODBC driver

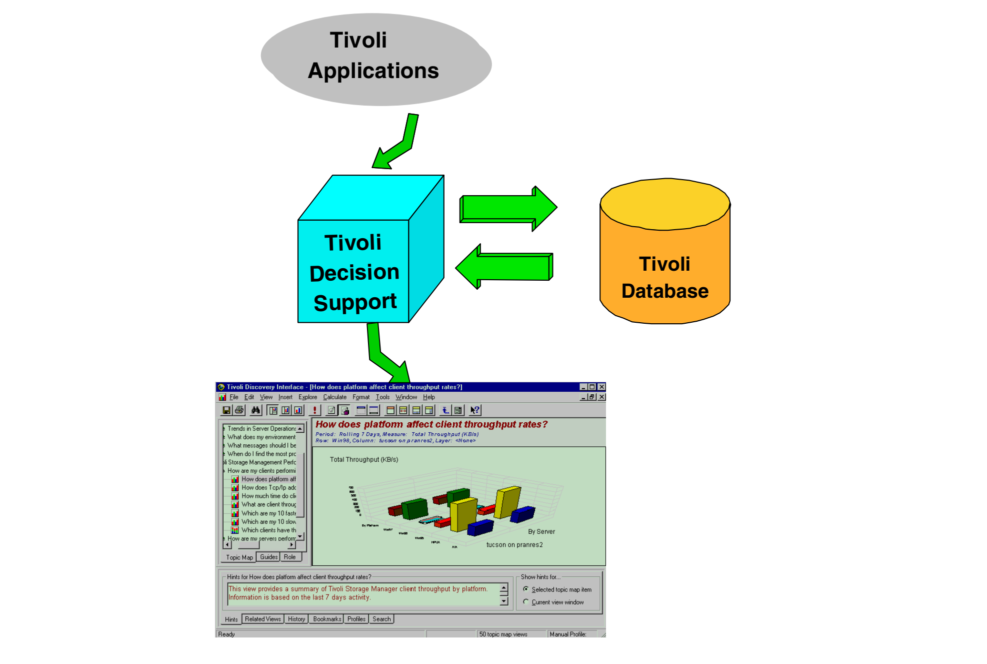
ODBC is a standard interface between SQL database engines and front-end applications. It allows products such as Lotus Approach or Microsoft Access to be used to graphically construct SQL queries, which are then dispatched to the database (in this case the Tivoli Storage Manager database). The select statement results are returned in tabular form and can be processed to be displayed as charts or tables. The ODBC driver is also a requirement for Tivoli Decision Support for Storage Management Analysis (see the next chapter). The Tivoli Storage Manager ODBC driver ships as part of the Win32 client package only. The driver is compliant with the ODBC 2.5 application programming interface (API).

30.6 Tivoli Storage Manager and Tivoli Decision Support

Along with the release of Tivoli storage Manager Version 3.7, Tivoli released a complementary product, Tivoli Decision Support for Storage Management Analysis. This provides tools and interfaces for the Tivoli Decision Support product to analyze and display data extracted from the Tivoli Storage Manager server. This data is used to obtain snapshot and exception reports, verify the trends for historical data and to forecast in the storage management environment. More information on this topic is available in the redbook *Tivoli Storage Manager Reporting*, SG24-6109.

30.6.1 Tivoli Decision Support

The product, Tivoli Decision Support (TDS) aids analysis and decision making in an organization. Figure 129 shows the overall structure of Tivoli Decision Support.



How does platform affect client throughput rates?

Period: Rolling 7 Days, Measure: Total Throughput (KB/s)
Row: Win98, Column: tucson on pranres2, Layer: <None>

Hints for How does platform affect client throughput rates?

This view provides a summary of Tivoli Storage Manager client throughput by platform. Information is based on the last 7 days activity.

Figure 129. Tivoli Decision Support overview

Tivoli Decision Support enables users to quickly find and use the data stored in the organization’s Tivoli databases. It extracts the gathered information and builds content automatically, ready for the users.

Tivoli Decision Support is a *decision-support* framework, made by components that aid in finding and analyzing the data.

By using Tivoli Decision Support to tap into the operational data an administrator can, for example:

- Measure the effectiveness of the operation.
- Determine how assets are distributed across the enterprise.
- Identify areas of weakness, to convert from reactive support activities to proactive support planning.
- Leverage investment in technology and automation.

The data can be viewed in a variety of formats and it is possible to view summary or detailed aspects, presented as text, bar chart, line chart or other graphical formats.

30.6.2 TDS components

TDS has two parts: the base product and the Decision Support Guides. The base product provides the basic functions to configure, administer and operate TDS. This is a prerequisite for using Decision Support guides. Decision Support Guides are software add-on modules for specific application support.

The components of TDS are shown in Figure 130.

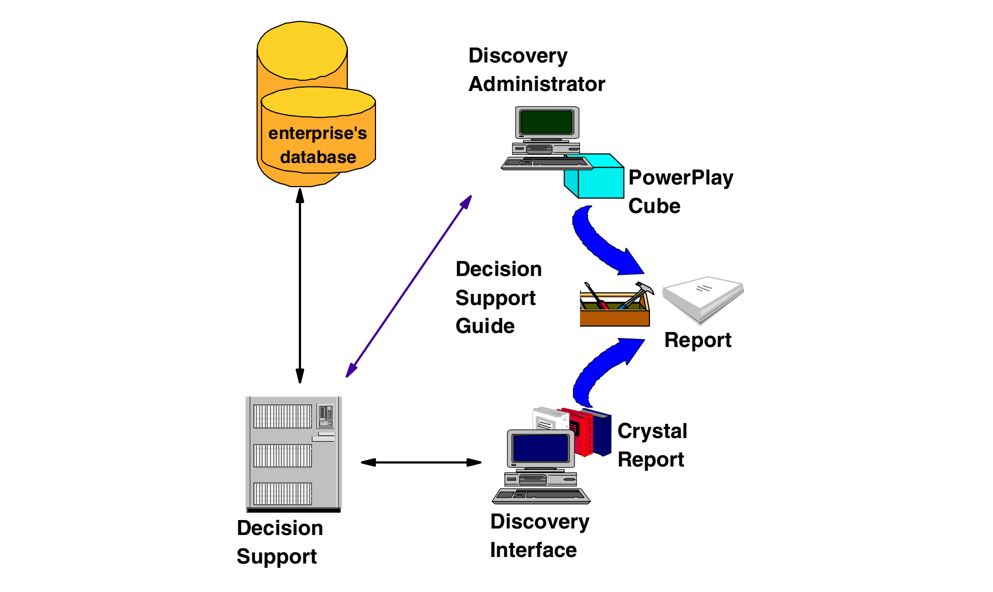


Figure 130. Tivoli Decision Support components

The components of the base product are:

- **Decision Support**, client-server framework software that runs in a Win2 environment, typically NT.

  The server component include the models, queries, templates and other information required to generate the views and acts as a central repository.

  The client component, also called **Tivoli Discovery Interface**, provides all the tools needed to open and work with views of data from the enterprise's database. This component must be installed on every system where TDS is used.

  Depending on the configuration, the server component could be installed in a single server machine, using a network model, or in all the machines where the client component is present, using a stand-alone mode.

- **Tivoli Discovery Administrator** is the module that is used to administer and configure TDS, schedule data generation processes in the enterprise's database and build customized cubes. A cube is a data container used by Cognos PowerPlay

- **Cognos PowerPlay** is a third-party application from Cognos. It generates multi-dimensional analyses and reports based on the historical data. PowerPlay can also generate customized views based on queries to the databases. The data is aggregated according to predefined models and the PowerPlay cube is built. The cube enables the data to be instantly viewed from many perspectives and in a variety of graphical styles including bar, line and pie charts. PowerPlay is installed on every system where TDS is used.

- **Crystal Reports** generates real-time text-based views in tabular form, using data extracted from standard queries. If text reports are required, then this component must be installed on the administrator's system.

Decision Support Guides are used by TDS to aid in discovering key information for applications or for a business concept, for example call center management. These pre-defined solutions provide direct answers to the most pertinent business questions through a simple question and topic paradigm and single views into the data. These solutions are created through knowledge of the data captured by Tivoli applications, knowledge of IT best practices, and knowledge of IT customer needs. They provide out-of-the-box data views and reports and are installed on the system where the TDS server is installed. The Decision Support Guides are used with the Tivoli Discovery Interface.

30.6.3 Tivoli Decision Support for Storage Management Analysis

The Tivoli Decision Support for Storage Management Analysis product provides "best practices" guides to analyze data and display reports for the following Tivoli Storage Manager main areas:

- Event Analysis
- Performance Analysis

The purpose of the event analysis guide is to gain insight into the health of a Tivoli Storage Manager environment, showing how well the environment is operating and what areas need improvement. The reports are generated for both Tivoli Storage Manager server and clients.

The purpose of the performance analysis guide is to gain insight into the environment's performance, showing how well the storage management environment is performing and what areas need improvement. Performance reports are also generated for both server and clients.

Future Storage Management Analyses guide releases will address other questions, particularly in the area of Tivoli Storage Management capacity planning.

A relational database is used to warehouse the Tivoli Storage Manager data, both historical and forecasted. The information can then be displayed in a variety of formats, graphical and text, and is possible to drill down from the summaries to get more detailed aspects of an environment.

TDS for Storage Management Analysis replaces Sterling's SAMS Vantage: ADSM edition product, which was used with ADSM V3.1 for reporting purposes.

The following section will discuss the architecture, main components and the data transfer model of Tivoli Decision Support for Storage Management Analysis.

30.6.4 TDS for Storage Management Analysis - Architecture

Figure 131 shows the layout of TDS for SMA. It consists of the following components:

1. **Storage Manager Decision Support Loader Tool** is a utility which uses the Tivoli Storage Manager ODBC driver to extract data from the server database and load it into the RDBMS for use. This module can be installed on any NT machine with a TCP/IP connection to both the Tivoli Storage Manager servers and the reporting database machine.
2. **RDBMS schemas** for the reporting database to create the data structure in the reporting database. The schemas must be loaded in the reporting database.
3. **Decision Support Guide for Storage Management Analysis** to provide the queries for Event and Performance reports. The guides must be imported into the Discovery Administrator.

Other components must also be present to make up a complete TDS for Storage Management Analysis reporting solution. These are:

- One or more Tivoli Storage Manager servers
- Tivoli Storage Manager / RDBMS interface
- A Relational Database System
- Tivoli Decision Support base product

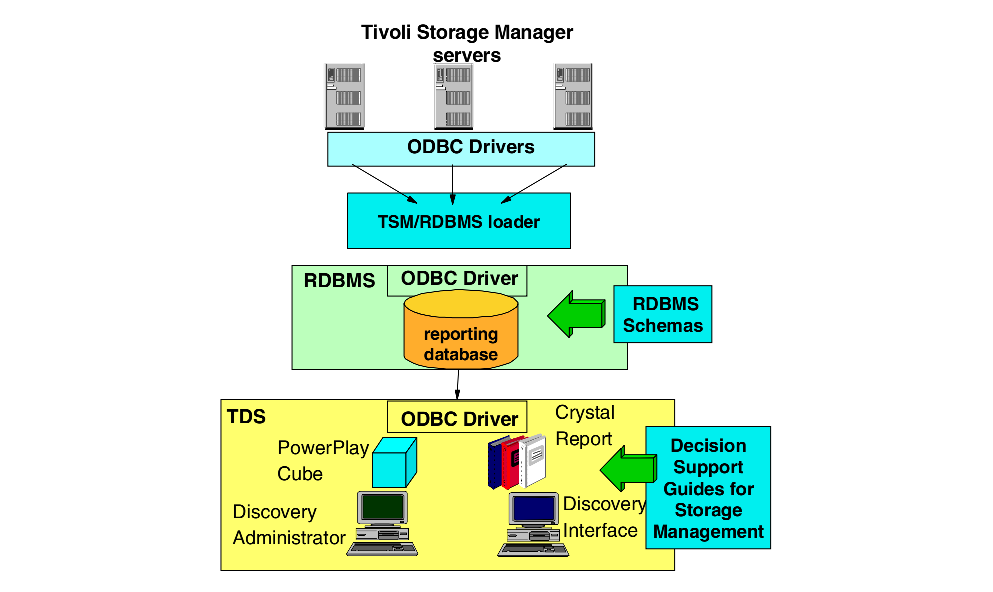


Figure 131. Architecture of TDS for Storage Management Analysis

30.6.4.1 Tivoli Storage Manager servers

The Tivoli Storage Manager database is the source of information used for event and performance information.

Version 3.7.2 or later of the Tivoli Storage Management server is required. This is because data is extracted from the Tivoli Storage Manager server from new tables included in this version. These are the SUMMARY and EVENTS tables. Therefore, this version is required to provide support for Storage Management Analysis information gathering.

There may be one or more Tivoli Storage Manager servers on the same or different operating system platforms. These are known to TDS for Storage Management Analysis as *source data servers*.

30.6.4.2 Tivoli Storage Manager / RDBMS interface

The major portion of this interface is the Decision Support Loader program. Its purpose is to move data from the Tivoli Storage Management source database to the Tivoli Decision Support reporting database. It is recommended to use a dedicated NT machine for the Decision Support Loader as it is a CPU-intensive process.

This program uses the Tivoli Storage Manager ODBC interface to read data from the Tivoli Storage Manager server and uses the RDBMS ODBC interface to save data into the Tivoli Storage Manager Reporting Database. The basic steps performed by the program during the data collection process are:

- Extract data from the Tivoli Storage Manager server using the Tivoli Storage Manager ODBC driver
- Transform data into the format required by the Tivoli Storage Manager reporting database. This includes providing a unique date/time stamp for the extracted information.
- Write data to the Tivoli Storage Manager Reporting database using the RDBMS ODBC driver.

The program also controls the expiration of old records from the RDBMS.

Data can be transferred concurrently from multiple Tivoli Storage Management servers to a shared RDBMS database server by installing the Decision Support Loader on multiple machines. To use this configuration, each Tivoli Storage Management server must be set up as an ODBC data source, and each Decision Support Loader must be configured to access the appropriate Tivoli Storage Management server and shared RDBMS database server.

Since the data must be extracted from Tivoli Storage Manager and into the TDS RDBMS, this means that the reports generated by TDS are not ’real time’ or ’live’. The reports will reflect data up to the point in time of the last data load process - this mode is known as ’near real-time’. So, there must be a trade-off between the desire to have current data for analysis and the overhead of running frequent data loads.

30.6.4.3 Relational Database System

The source information is saved in a relational database called the Tivoli Storage Manager reporting database. One Tivoli Storage Manager reporting database can be used to contain data from multiple Tivoli Storage Manager servers. The Tivoli Storage Manager reporting databases that are initially supported are:

- Oracle (V8.1.5)
- DB2(V5.2)
- MS SQL(v7.0)

A schema is provided with Tivoli Decision Support for Storage Management Analysis that contains the definitions of the tables used in the RDBMS for these three.

30.6.4.4 Tivoli Decision Support base product

The Tivoli Decision Support base product includes several components as already explained in section 30.6.2, “TDS components” on page 283. There are many possible configurations for TDS, including which components will be installed, and which machines will run them. TDS can be run in stand-alone or network mode.

Tivoli Decision Support operates on Windows 95/98 and Windows NT 4. Note that Tivoli Decision Support does not pre-requisite the Tivoli Framework.There may be multiple Tivoli Decision Support servers in the same reporting system. TDS must be at least Version 2.1 with PTF2 in order to support TDS for Storage Management Analysis. More information on configurations and hardware requirements is available in *Tivoli Decision Support Installation Guide,* GC32-0438.

30.6.4.5 TDS for Storage Management Analysis components

The Storage Management Analysis Guide delivered with the Tivoli Decision Support for Storage Management Analysis product consists of:

- **“Out-of-box” reports:** This sets up the reporting links to the reporting data base, Cognos, and Crystal. This is done via Tivoli Decision Support and its GUI interface that give access to these reports with multiple viewing options.
- **Cube reporting:** This component sets up reporting from Cognos 3D graphics reporting.

It supplies comprehensive pre-coded reports which are designed to meet most, if not all reporting requirements. In fact, the TDS Guide for Storage Management Analysis is specifically ’not’ intended to be a customizable solution. The complexity of this six product solution (Tivoli Storage Manager, TDS, Guide for Storage Management Analysis, Cognos PowerPlay and Seagate Crystal Reporting) allows little room for custom-built reports and would introduce high risk if attempted. However, the next section will show a sample of the many in-built reports and queries provided.

Tivoli Decision Support for Storage Management Analysis requires a Windows NT 4.0 SP4 level or higher system.

30.6.5 Reporting example using TDS for SMA

The Tivoli Storage Management administrator should setup a regularly executed schedule for loading the reporting database and cube building. Dataloads should be scheduled to avoid times when the Tivoli Storage Manager database is being heavily used. Avoiding times when client backups and archives are running is preferred. Avoid running dataloads during Tivoli Storage Manager administrative jobs. Expect degraded performance if this is not possible.

Data must be loaded into the Cognos cube from the TDS RDBMS. This is a periodic job which should be scheduled after the dataloader process is completed.

Reporting using System Management Analysis should be done on a regularly scheduled basis, as well. This will make trends and anomalies more apparent to the viewer and increase the value of the product. Specific reports to be generated on a weekly, monthly, daily, or weekend basis are dependent on individual preference and operational relevance; however a very large number of reports are provided.

The following section is a recommended guide for a Storage Management Analysis report schedule.

30.6.5.1 Daily reporting

Having completed the Tivoli Storage Manager Decision Support Loader and the cube builds scheduling, the reports can be viewed in the Tivoli Discovery Interface. These reports can show possible server or client errors that should be checked daily. Use the following reports from the Tivoli Discovery Interface for daily reporting:

Tivoli Storage Management Event Analysis

1. Daily exception reports
   - Classify outcome for client sessions
   - Summarize the messages issued in my environment
   - What are the results of all operations that occurred for a client
   - What is the status of non-scheduled administrative operations
   - What is the status of scheduled administrative operations
   - What is the status of scheduled client sessions
2. How might I improve my storage health
   - How current are client backups
   - Show me my server database backup record
   - When did clients last contact the server
3. Trends in Server Operations
   - What is the average daily failed object count
   - What is the average daily successful object count
4. What messages should I be concerned with
   - Summarize client messages by severity
   - Summarize server error messages by severity
   - What are the top 10 server messages

Tivoli Storage Management Performance Analysis

1. How are my clients performing
   - Which are my 10 fastest clients
   - Which are my 10 slowest clients
2. How does time of day and day of week affect my performance
   - What is the average daily throughput for client operations
   - What is the average hourly throughput for client operations
   - What is the average hourly throughput take for server operations

This set of reports should point to immediate problems which need attention. Unexplained ‘spikes’ in the performance and object failure are points for further investigation, as are messages indicating severe and critical errors.

30.6.5.2 Weekly reporting

Use the following reports from the Tivoli Discovery Interface for weekly reporting:

Tivoli Storage Management Event Analysis

1. How might I improve my storage health
   - How are clients schedules affected by the day of the week
   - How current are client backups
   - What administrators and clients have been locked out of the system
2. Trend in Client Operations
   - What is the average daily failed object count
   - What time of day do peak client data transfers occur
3. Trends in Server Operations
   - How does time affect server session activity
   - What is the daily failed object count for server operations
4. What messages should I be concerned with
   - Summarize client messages by severity
   - What are the top 10 client messages with the most problems
5. When do I find the most problems
   - What day of week do peak client error rates occur
   - What day of week do peak server error rates occur
   - What time of day do peak clients error rates occur
   - What time of day do peak server error rates occur

Tivoli Storage Management Performance Analysis

1. How are my clients performing
   - How does TCP/IP address affect throughput

These reports should alert the administrator to potential problems, unusual growth which could lead to problems, or beginning trends.

30.6.5.3 Weekend reporting

The following reports from the Tivoli Discovery Interface are recommended for weekend reporting:

Tivoli Storage Management Event Analysis

1. When do I find the most problems
   - What day of week do peak client error rates occur
   - What day of week do peak server error rates occur

Tivoli Storage Management Performance Analysis

1. How are my servers performing
   - How much time is spent daily waiting for media access
2. How does file size affect my performance
   - How does file size affect client operation throughput
3. How does time of day and day of week affect my performance
   - What is the average daily throughput for client operations
   - What is the average hourly throughput for client operations
   - What is the average hourly throughput take for server operations

Weekend reporting is much like weekly in purpose. Some organizations prefer to look at data as it corresponds to the calendar week. You may find that this is appropriate. If not, run these along with the weekly reports.

30.6.5.4 Monthly reporting

The following from the Tivoli Discovery Interface are recommended for monthly reporting:

Tivoli Storage Management Event Analysis

1. Trend in Client Operations
   - Summarize client messages issued by platform
   - Summarize client messages issued by software level
   - What time of day do peak client data transfers occur
   - Which clients have the highest average failed objects count
2. Trends in Server Operations
   - How does time affect server session activity
   - How much data is processed by daily server operations
   - What is the daily failed object count for server operations
3. What does my environment look like
   - How many clients are registered with each server
4. What messages should I be concerned with
   - Summarize client messages by severity
   - What are the top 10 clients with the most problems

Tivoli Storage Management Performance Analysis

1. How are my clients performing
   - How does platform affect client throughput rates
   - How does TCP/IP address affect throughput
2. How does file size affect my performance
   - Classify sessions by file size and performance thresholds
   - How does file size affect client operation throughput

These reports, generated with the proper date filters, will yield an analysis of performance and events within the month. They will give a picture of what has transpired during the month, as well as show storage trends in the enterprise. These may be good candidates to be stored for future reference or year end summary reporting.

Appendix A. Special notices

This publication is intended to help network managers and IT Consultants to understand the features and functions of Tivoli Storage Manager. It introduces Tivoli Storage Manager concepts for those new to storage management, in general, and Tivoli Storage Manager, in particular.

The information in this publication is not intended as the specification of any programming interfaces that are provided by Tivoli Storage Manager. See the PUBLICATIONS section of the IBM Programming Announcement for Tivoli Storage Manager for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

ADSTAR
AIX
AnyNet
APPN
AS/400
AT
CT
DataPropagator
DB2
IBM
ImagePlus
IMS
MVS/ESA
Netfinity
NetView
OfficeVision
Operating System/2
Operating System/400
OS/2
OS/390
OS/400
RETAIN
RS/6000
S/390
SP
SP1
SP2
System/390
SystemView
VisualGen
VisualInfo
XT
400

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

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

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and/or other countries licensed exclusively through X/Open Company Limited.

SET and the SET logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.

Appendix B. Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

B.1 IBM Redbooks

For information on ordering these ITSO publications see “How to get IBM Redbooks” on page 297.

.

Book Title	**Publication Number**
General Topics	
Getting Started with Tivoli Storage Manager: Implementation Guide	SG24-5416
Tivoli Storage Manager Version 3.7: Technical Guide	SG24-5477
Using ADSM Hierarchical Storage Management	SG24-4631
Client Disaster Recovery: Bare Metal Restore	SG24-4880
Planning for IBM Remote Copy	SG24-2595
P/DAS and Enhancement to the IBM 3990-6 and RAMAC Array Family	SG24-4724
TME 10 Cookbook for AIX: Systems Management and Networking Applications	SG24-4867
An Introduction to Tivoli Enterprise	SG24-5494
Implementing ESS Copy Services on System 390	SG24-5680
Implementing ESS Copy Services on UNIX and Windows NT	SG24-5757
Slicing the AS/400 with Logical Partitioning: a How to Guide	SG24-5439
Specific Server Books	
A Practical Guide to Implementing Tivoli Storage Manager on AS/400	SG24-5472
Windows NT Backup and Recovery with ADSM	SG24-2231
Using TSM in a Clustered NT Environment	SG24-5742
ADSM Server-to-Server Implementation and Operation	SG24-5244
ADSM Server for Windows NT Configuration and Recovery Examples	SG24-4878
ADSM/6000 on 9076 SP2	GG24-4499
ADSM for OS/2 Advanced Topics	SG24-4740
Tivoli Storage Manager with Other Products	
A Practical Guide to Network Storage Manager	SG24-2242
Using ADSM to Back Up Databases	SG24-4335
Using Tivoli Storage Manager to Back Up Lotus Domino R5	SG24-5247
Using Tivoli Storage Manager to Back Up Lotus Notes	SG24-4534
SAP R/3 Data Management with Tivoli Storage Manager	SG24-5743
Using ADSM to Back Up and Recover Microsoft Exchange Server	SG24-5266
Using ADSM to Back Up OS/2 LAN Server and Warp Server	SG24-4682
Backup, Recovery, and Availability with DB2 Parallel Edition on RISC/6000,	SG24-4695
ADSM Operation and Management with TME10,	SG24-2214
Tivoli Storage Management Reporting	SG24-6109
Using Tivoli Data Protection for Workgroups,	SG24-5490

B.2 IBM Redbooks collections

Redbooks are also available on the following CD-ROMs. Click the CD-ROMs button at ibm.com/redbooks/ for information about all the CD-ROMs offered, updates and formats.

CD-ROM Title	Collection Kit Number
System/390 Redbooks Collection	SK2T-2177
Networking and Systems Management Redbooks Collection	SK2T-6022
Transaction Processing and Data Management Redbooks Collection	SK2T-8038
Lotus Redbooks Collection	SK2T-8039
Tivoli Redbooks Collection	SK2T-8044
AS/400 Redbooks Collection	SK2T-2849
Netfinity Hardware and Software Redbooks Collection	SK2T-8046
RS/6000 Redbooks Collection (BkMgr Format)	SK2T-8040
RS/6000 Redbooks Collection (PDF Format)	SK2T-8043
Application Development Redbooks Collection	SK2T-8037
IBM Enterprise Storage and Systems Management Solutions	SK3T-3694

B.3 Tivoli Storage Management Publications

Book Title	Publication Number
Tivoli Storage Manager for AIX Version 3.7 Quick Start	GC35-0367
Tivoli Storage Manager for AIX Version 3.7 Administrator’s Guide	GC35-0368
Tivoli Storage Manager for AIX Version 3.7 Administrator’s Reference	GC35-0369
Tivoli Storage Manager for HP-UX Version 3.7 Quick Start	GC35-0370
Tivoli Storage Manager for HP-UX Version 3.7 Administrator’s Guide	GC35-0371
Tivoli Storage Manager for HP-UX Version 3.7 Administrator’s Reference	GC35-0372
Tivoli Storage Manager for Sun Solaris Version 3.7 Quick Start	GC35-0373
Tivoli Storage Manager for Sun Solaris Version 3.7 Administrator’s Guide	GC35-0374
Tivoli Storage Manager for Sun Solaris Version 3.7 Administrator’s Reference	GC35-0375
Tivoli Storage Manager for MVS and OS/390 Version 3.7 Quick Start	GC35-0276
Tivoli Storage Manager for MVS and OS/390 Version 3.7 Administrator’s Guide	GC35-0277
Tivoli Storage Manager for MVS and OS/390 Version 3.7 Administrator’s Reference	GC35-0278
Tivoli Storage Manager for Windows NT Version 3.7 Quick Start	GC35-0379
Tivoli Storage Manager for Windows NT Version 3.7 Administrator’s Guide	GC35-0380
Tivoli Storage Manager for Windows NT Version 3.7 Administrator’s Reference	GC35-0381
Tivoli Storage Manager Version 3.7 Messages	GC35-0382
Tivoli Storage Manager for AS/400 Version 3.1.2 Quick Start	GC35-0317
Tivoli Storage Manager for AS/400 Version 3.1.2 Administrator’s Guide	GC35-0315
Tivoli Storage Manager for AS/400 Version 3.1.2 Administrator’s Reference	GC35-0316
Tivoli Storage Manager for VM Version 3.1 Quick Start	GC35-0351
Tivoli Storage Manager for VM Version 3.1 Administrator’s Guide	GC35-0352
Tivoli Storage Manager for VM Version 3.1 Administrator’s Reference	GC35-0353
Tivoli Storage Manager for VM Version 3.1 Messages	GC35-0271
Tivoli Storage Manager for OS/2 Version 2.1 Quick Start	GC35-0231
Tivoli Storage Manager for OS/2 Version 2.1 Administrator’s Guide	GC35-0232
Tivoli Storage Manager for OS/2 Version 2.1 Administrator’s Reference	GC35-0233
Tivoli Storage Manager for OS/2 Version 2.1 Messages	SH35-0133
Tivoli Storage Manager Version 3.7 Using the Backup-Archive Client for UNIX	SH26-4105

Book Title	**Publication Number**
Tivoli Storage Manager Version 3.7 Using the Backup-Archive Client for Windows	SH26-4101
Tivoli Storage Manager Version 3.7 Using the Backup-Archive Client for NetWare	SH26-4100
Tivoli Storage Manager Version 3.7 Installing the Clients	SH26-4102
Tivoli Storage Manager Version 3.7 Using the Application Programming Interface	SH26-4107
Tivoli Storage Manager Version 3.7 AFS/DFS Backup Clients	SH26-4106
Tivoli Storage Manager Version 3.7 Trace Facility Guide	SH26-4104
Tivoli Storage Manager Version 3.7 Reference Cards for the Backup-Archive Clients	SX26-6021
Tivoli Data Protection for Informix V3R7: Installation and User’s Guide	SH26-4095
Tivoli Data Protection for Lotus Domino for UNIX V1R1: Installation and User’s Guide	SH26-4088
Tivoli Data Protection for Lotus Domino for Windows NT V1R1: Installation and User’s Guide	GC26-7320
ADSMConnect Agent for Lotus Notes on Windows NT Installation and User’s Guide	SH26-4065
ADSMConnect Agent for Lotus Notes on AIX Installation and User’s Guide	SH26-4067
Tivoli Data Protection for Microsoft Exchange Server Installation and User’s Guide	SH26-4071
Tivoli Data Protection for Microsoft SQL Server Installation and User’s Guide	SH26-4069
Tivoli Data Protection for Oracle Backup on AIX Installation and User’s Guide	SH26-4061
Tivoli Data Protection for Oracle Backup on Sun Solaris Installation and User’s Guide	SH26-4063
Tivoli Data Protection for Oracle Backup on HP-UX Installation and User’s Guide	SH26-4073
Tivoli Data Protection for Oracle Backup on Windows NT Installation and User’s Guide	SH26-4086
Tivoli Data Protection for Microsoft SQL Server’s Installation Guide	SH26-4111

Additional Titles

IBM Enterprise Storage Server Introduction and Planning Guide	GC26-7294
Tivoli Decision Support Installation Guide	GC32-0328

How to get IBM Redbooks

This section explains how both customers and IBM employees can find out about IBM Redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site** ibm.com/redbooks

  Search for, view, download, or order hardcopy/CD-ROM Redbooks from the Redbooks Web site. Also read redpieces and download additional materials (code samples or diskette/CD-ROM images) from this Redbooks site.

  Redpieces are Redbooks in progress; not all Redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

- **E-mail Orders**

  Send orders by e-mail including information from the IBM Redbooks fax order form to:

	e-mail address
In United States or Canada	pubscan@us.ibm.com
Outside North America	Contact information is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	Country coordinator phone number is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Fax Orders**

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	Fax phone number is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

This information was current at the time of publication, but is continually subject to change. The latest information may be found at the Redbooks Web site.

IBM Intranet for Employees

IBM employees may register for information on workshops, residencies, and Redbooks by accessing the IBM Intranet Web site at http://w3.itso.ibm.com/ and clicking the ITSO Mailing List button. Look in the Materials repository for workshops, presentations, papers, and Web pages developed and written by the ITSO technical professionals; click the Additional Materials button. Employees may access MyNews at http://w3.ibm.com/ for redbook, residency, and workshop announcements.

IBM Redbooks fax order form

Please send me the following:

Title	Order Number	Quantity

First name　Last name

Company

Address

City　Postal code　Country

Telephone number　Telefax number　VAT number

□ Invoice to customer number

□ Credit card number

Credit card expiration date　Card issued to　Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

Index

IBM Redbooks review

Your feedback is valued by the Redbook authors. In particular we are interested in situations where a Redbook "made the difference" in a task or problem you encountered. Using one of the following methods, **please review the Redbook, addressing value, subject matter, structure, depth and quality as appropriate.**

- Use the online **Contact us** review redbook form found at ibm.com/redbooks
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Document Number **Redbook Title**	SG24-4877-02 Tivoli Storage Management Concepts
Review	
What other subjects would you like to see IBM Redbooks address?	
Please rate your overall satisfaction:	O Very Good O Good O Average O Poor
Please identify yourself as belonging to one of the following groups:	O Customer O Business Partner O Solution Developer O IBM, Lotus or Tivoli Employee O None of the above
Your email address: The data you provide here may be used to provide you with information from IBM or our business partners about our products, services or activities.	 O Please do not use the information collected here for future marketing or promotional contacts or other communications beyond the scope of this transaction.
Questions about IBM's privacy policy?	The following link explains how we protect your personal information. ibm.com/privacy/yourprivacy/

IBM

Redbooks

Tivoli Storage Management Concepts

(0.5” spine)
0.475”<->0.873”
250 <-> 459 pages

Tivoli Storage Management Concepts

Ideal introduction to all aspects of Tivoli Storage Management

Learn how to protect your vital applications and data

Essential reading for all systems administrators!

Tivoli Storage Management solutions provide enterprise wide distributed storage management. This IBM Redbook describes the features and functions of Tivoli Storage Manager, the backbone of the Tivoli Storage Management product set. It introduces Tivoli Storage Management concepts for those new to storage management, in general, and to Tivoli Storage Manager, in particular.

This redbook gives a broad understanding of the architecture of Tivoli Storage Manager, and will help you design solutions to protect your data assets from losses ranging in scale from those caused by user error through complete site disasters.

The book also describes the integration of Tivoli Storage Manager with many popular applications, such as databases and groupware. This includes a discussion of other products in the Tivoli Storage Management group, including Tivoli Data Protection application agents, as well as other complementary products from IBM, Tivoli and other vendors.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-4877-02 ISBN 0738419222