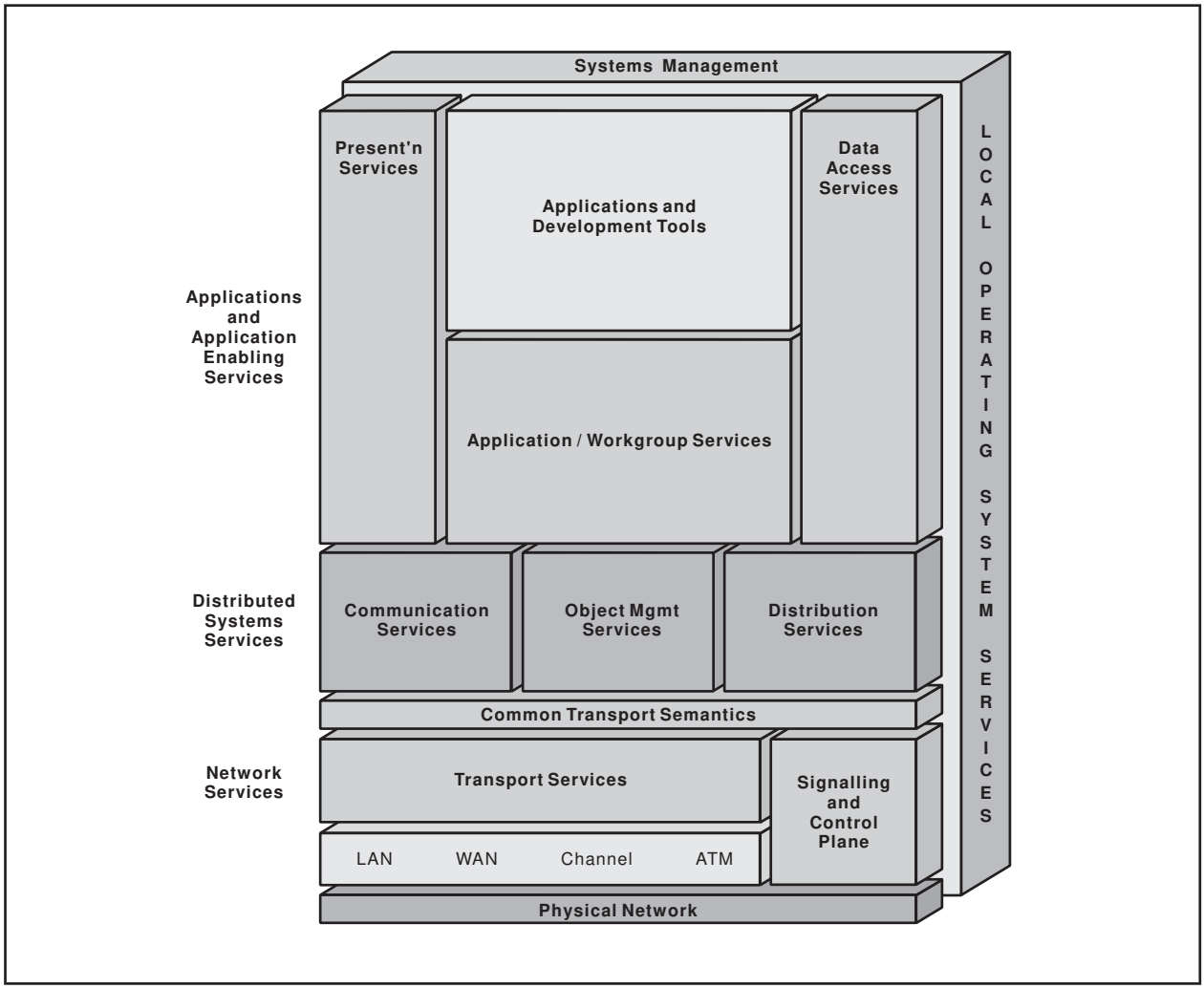
Open Blueprint

Storage Management Resource Manager



GC23-3927-01

Open Blueprint

Storage Management Resource Manager

GC23-3927-01

About This Paper

Open, distributed computing of all forms, including client/server and network computing, is the model that is driving the rapid evolution of information technology today. The Open Blueprint structure is IBM's industry-leading architectural framework for distributed computing in a multivendor, heterogeneous environment. This paper describes the Storage Management resource manager component of the Open Blueprint and its relationships with other Open Blueprint components.

The Open Blueprint structure continues to accommodate advances in technology and incorporate emerging standards and protocols as information technology needs and capabilities evolve. For example, the structure now incorporates digital library, object-oriented and mobile technologies, and support for internet-enabled applications. Thus, this document is a snapshot at a particular point in time. The Open Blueprint structure will continue to evolve as new technologies emerge.

This paper is one in a series of papers available in the *Open Blueprint Technical Reference Library* collection, SBOF-8702 (hardcopy) or SK2T-2478 (CD-ROM). The intent of this technical library is to provide detailed information about each Open Blueprint component. The authors of these papers are the developers and designers directly responsible for the components, so you might observe differences in style, scope, and format between this paper and others.

Readers who are less familiar with a particular component can refer to the referenced materials to gain basic background knowledge not included in the papers. For a general technical overview of the Open Blueprint, see the *Open Blueprint Technical Overview*, GC23-3808.

Who Should Read This Paper

This paper is intended for audiences requiring technical detail about the Storage Management Resource Manager in the Open Blueprint. These include:

- Customers who are planning technology or architecture investments
- Software vendors who are developing products to interoperate with other products that support the Open Blueprint
- Consultants and service providers who offer integration services to customers

© **Copyright International Business Machines Corporation 1995, 1996. All rights reserved.**
Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Summary of Changes 1

Storage Management Resource Manager 3
Major Components of the Storage Management Resource Manager 3
Transparent Functions of the Storage Management Resource Manager 4
Storage Management Resource Manager API 5
Network Computing 8
Mobile Computing 8
Characteristics of the Storage Management Resource Manager 8
Relationship to Other Resource Managers 10
Summary 10

Appendix A. Bibliography 11

Appendix B. Notices 13
Trademarks 13

Appendix C. Communicating Your Comments to IBM 15

© Copyright IBM Corp. 1995, 1996

Summary of Changes

This revision includes information about:

- Additional support for scalability
- Support for network and mobile computing
- The relationship with the HTTP resource manager

© Copyright IBM Corp. 1995, 1996

Storage Management Resource Manager

The Storage Management resource manager provides cost-effective management and protection of data in a heterogeneous, multi-system environment. These facilities are provided both transparently and through an application programming interface (API). Storage management components operate in a client/server model where the client performs the storage management operations for the local system, and the server provides storage facilities and centralized management for all the clients in an enterprise.

This paper describes the main management and system components of the Storage Management resource manager and how they are used within an installation. The paper also describes the relationships to other resource managers in the Open Blueprint structure.

Storage management consists of the following four elements:

Management Class Defines storage availability, space, and retention attributes. These attributes define policies that manage:

- Backup data (number of versions, and so on)
- Archive data (retention periods, and so on)
- Space management data (migrate and recall functions)

Data Class Specifies the definition of the data when new files are created. For example, the file can be defined as stream-oriented, record-oriented, or another format as required by the system.

Storage Class Defines the performance and availability requirements of a file. The Storage Management resource manager will place the file (or data) on the device that most closely meets its storage class requirements. Placement can be on optical media or high-speed, high-availability RAID storage.

Storage Group Defines the set of storage volumes for future allocation. When new data is created, it can be placed into one of the available storage groups that offer predefined management attributes.

Collectively, these storage categories define the storage management policies for data on the system.

Major Components of the Storage Management Resource Manager

The major components that define and carry out the policies defined in the previous categories are:

- **Server**. A component which allows a system to act as a storage management server for workstation and file server clients. The server provides all tracking and management of data stored in the system. It provides hierarchical storage management functions by linking together multiple storage pools (sets of like storage devices or volumes) into a storage hierarchy based on performance and availability characteristics. Data is placed within the hierarchy as directed by the storage management policies.
- **Administrative**. An application that allows an administrator to control and monitor server and client activities, define storage management policies, and set up schedules for operations. This component can be run from anywhere in the network to manage the system. The administrative application operates with other Open Blueprint Systems Management services for seamless control and operation across an enterprise.
- **Backup-Archive**. A client component that allows users to back up and archive files from their workstations or file servers to server storage, and restore and retrieve files to their local file system.

© Copyright IBM Corp. 1995, 1996

- **Hierarchical Storage Management (HSM)**. A client component that provides space-management function to workstations. HSM automatically migrates eligible files to server storage to maintain specific levels of free space on the local file system. It automatically recalls migrated files on access.
- **Application Programming Interface (API)**. An API that is based on the emerging X/Open standard interface for pro-active management and protection of data. The requirements of some application areas and resource managers, such as databases and multimedia, are not always met by file systems. For these, the Storage Management resource manager provides the ability to directly manage the archiving and backup of data.

These components can all operate in a highly-distributed environment, or can all operate on a single system. For example, the server can be an IBM RS/6000 running AIX with a storage hierarchy of disk and robotic tape units. The administrator can run from a PC running OS/2. Backup, archive, and HSM clients can operate on Novell Netware, Windows, Sun, AS/400, and other platforms. The configuration is left to users to best meet their needs.

Transparent Functions of the Storage Management Resource Manager

The Storage Management resource manager offers a wide set of transparent functions to applications. These functions allow applications to operate and take advantage of storage management operations without modification.

The transparent functions are typically provided through a high level of integration with the File resource manager. If an application uses the file system, its data can be fully managed transparently through the Storage Management resource manager.

The transparent functions are:

- Data backup and restore
- Data archive and retrieve
- Migrate and recall functions through hierarchical storage management (HSM) facilities
- Data allocation

Data Backup and Archiving

The Storage Management resource manager provides file protection using state of the art backup systems and data archiving (long term file and named binary object preservation). Backup and archive differ in their storage management policies, but are exactly the same in their interaction with the File resource manager.

Backup and archive perform operations on the file systems (or the underlying storage media) in parallel with standard application access. These operations are typically scheduled, but can be initiated by system events or in an ad hoc manner. System administrators can dictate through storage management policies whether exclusive access is required for a file to be backed up or archived, or whether shared access is permitted.

The backup and archive facilities are provided transparently to most applications on the system. If the application uses the file system in standard, supported ways, no action is required on the part of the application to receive these services. If specialized support is required, the storage management API should be used.

Space Management Through Migrate and Recall

Space management is the process used to keep space available in the local file systems by migrating data that is less frequently accessed to the storage management server. When accessed, files are transparently recalled to the local file system for standard access. Storage management policies indicate how much free space to maintain on the local system, which files are candidates for migration, whether backup copies should also exist for migrated data, and what storage hierarchy the migrated data should be placed in.

Components of space management, unlike backup and archive, must operate in conjunction with the File resource manager. These components allow space management to intercept file activities such as file open and delete. File system notifications such as out of space conditions must also be intercepted so that the space management component can free up space for standard operations to continue.

The space management operations are transparent to any application, just as the backup and archive operations are. System administrators can tune the operation of the space management facilities to best meet the needs of the applications on the system. For example, a data analysis application that needs to collect some statistics on data collected for the previous month reads all data files once in a sequential manner, and no updates are made. These applications run in a mode called *read without recall*, where any data that has been migrated is not recalled to local storage, but instead is recalled into the memory buffers that the application is using. Thus, the data can be accessed as if it had been resident on local storage; applications can operate in a more efficient mode without disrupting other applications using the local file system concurrently.

Data Allocation Services

The storage management backup, archive, and space management functions all operate on data that is already on the system. This requires that a system administrator set up a system with the proper storage resources, and define the policies to manage those resources. The Storage Management resource manager provides data allocation services that help automate these operations for a system.

Data allocation services, like space management, have key components that are tightly integrated with the File resource manager. These services provide for automatic creation and placement of new file systems to meet the storage management policies as defined by the system administrator. The data allocation services are integrated with the data allocation function of the File resource manager.

Storage Management Resource Manager API

The Storage Management resource manager provides a distributed API that provides a source-level interface for applications or facilities needing managed storage of named binary objects. The interface is platform-independent, vendor-independent, and application-independent, and is similar to standard file I/O interfaces.

Applications or services using the API are clients to a storage management server on which the named binary objects are stored and managed. The API consists of calling interfaces, structures, and return codes. It enables applications or services to back up and restore data objects, to archive and retrieve data objects, and to query or delete data on the server. The calls are independent of the application's user interfaces, transfer protocols, and media.

Applications that do not make standard use of the file system or that have unique storage management requirements, use the storage management API to fulfill their storage requirements. Examples of these applications are:

Database Databases must remain available a high percentage of the time, and many must be available seven days a week, 24 hours a day (7x24). For these systems, halting the database for back up is unacceptable. Also, databases typically store large amounts of data in large files, or on what are known as raw volumes (logical storage volumes without an underlying file system). Because typical database operations only modify a small percentage of this data daily, only the changed data needs to be backed up on a regular basis. Because each database is different, it is impossible for the storage management facilities to determine what has changed to perform an efficient backup. For this reason, databases use the storage management API to back up their data. Because the database knows what has changed, it can perform the backup while staying online. This integration meets the 7x24 availability requirement, and meets the requirement for central management of storage resources.

Multimedia Multimedia systems must store large amounts of data in an efficient manner. Typically, they also have multiple files that make up a full multimedia presentation (the presentation script, video data, and audio data). The multimedia application needs these files stored in an appropriate grouping to ensure proper access when required. For example, the Digital Library resource manager uses the storage management API to store data components in the storage server until access to them is required. As required, the Digital Library resource manager recalls all data components needed to fulfill a user's request. Because only the digital library knows the proper relationships between the data components, it must be involved in the access to those components.

There are many other examples, but most applications and resource managers that use storage management either have very large storage requirements or have special data relationships that must be maintained.

Storage Management Resource Manager API Structure

The Storage Management resource manager API is composed of the following major components:

- Session setup, initialization, and termination
- Transaction services
- Data creation, get, put, and deletion operations
- Application of storage management policies
- Data query operations

All API operations that create, store, update, or delete objects in the system are performed within atomic transactions. These transactions ensure the consistency of the system, and where any transaction is aborted, all updates are rolled back to the state where the transaction was begun. Applications need not worry about writing incomplete data to the storage server or how to provide cleanup operations should a transaction need to be aborted. Both the storage server and the API application can choose to abort a transaction. Multiple objects can also be sent in a single transaction to enhance overall system performance.

The API's components and corresponding API calls are listed below.

Session APIs: This set of APIs provides session service and maintenance routines that apply to the session layer. The session API calls are:

dsmQueryApiVersion Query the version of the API

dsmInit Initialize a communication session with the server

dsmQuerySessInfo Query information about the communication session established with the server

dsmTerminate Terminate a communication session with the server

Transaction APIs: This set of APIs allows applications to define the beginning and end of a transaction for write operations to the storage management server. The transaction API calls are:

dsmBeginTxn Begin a transaction with the server

dsmEndTxn End a transaction with the server

Data Handling APIs: This set of APIs provide for the actual data handling operations between the API client and the storage server. The data handling API calls are:

dsmRegisterFS Register a filespace with the server

dsmDeleteFS Delete a filespace on the server

dsmUpdateFS Update filespace information on the server

dsmGetObj Initiate a request for restoring or retrieving a copy of a specific data object from the server

dsmBeginGetData Begin a restore or retrieve operation

dsmGetData Get a byte stream of data from the server. This call is used in conjunction with the **dsmGetObj** call

dsmEndGetData End a restore or retrieve operation

dsmEndGetObj End the cycle for restoring or retrieving a copy of a specific data object from the server

dsmSendObj Initiate a request to send a data object to the server for backup or archive purposes

dsmSendData Send a byte stream of data to the server. This call is used in conjunction with the **dsmSendObj** call

dsmEndSendObj End the cycle for sending data for a specific data object to the server

Applying Storage Management Policies: This API call associates, or binds, the storage management policies to the object being created. The storage management API call is:

dsmBindMC Bind policy information to a data object

Data Query Operations: This set of APIs allows the calling application to query the storage management server for the named binary objects and their meta data. The data query API calls are:

dsmBeginQuery Initiate a query request for backup, archive, filespace, or other type of data from the server

dsmGetNextQObj Get the next query response from the server

dsmEndQuery End a query request for data from the server

Standards

The storage management API conforms to the proposed standards documented in the technical paper, *A Model for Backup, Archive, and Restore: A Proposal to X/Open* November 1993, Order 94A 00127, Report RJ-9620. This report is the basis for the X/Open preliminary specification, *Systems Management: Backup Services API (XBSA).*

Network Computing

The Storage Management resource manager supports the Open Blueprint Common Transport Semantics, and thus supports the base connections that are widely used for network computing. This support allows the Storage Management resource manager to operate seamlessly over the Internet from any of the supported platforms. The Storage Management resource manager supports a World Wide Web interface into the system by using the standard HTTP resource manager CGI interface for administrative sessions. Administrators can use standard Web browsers to manage the Storage Management resource manager. This Web support can exploit secure HTTP protocols, which provide a higher level of security.

Sample Web applications are also delivered with the Storage Management resource manager that demonstrate how installations can further exploit the Web to perform operations on remote systems, for example backup or restore.

Mobile Computing

The Storage Management resource manager supports the mobile user. Key characteristics of mobile users are long periods of disconnected operation and relatively slow connections to the home site when compared to local LAN connections. The Storage Management resource manager uses a form of incremental backup called *progressive incremental*. Using this method after the initial backup, only new or changed data needs to be backed up. The Storage Management resource manager must be notified of which data files have been deleted since the last backup. Progressive incremental backup dramatically reduces the amount of data that is sent over a network from a mobile user. This function, coupled with data compression and file filtering to exclude common files, gives the mobile user an efficient solution for managing and protecting their storage whether at home or on the road.

Characteristics of the Storage Management Resource Manager

Performance

Storage management systems must perform at acceptable rates to meet customers' data availability requirements. These rates are typically defined outside of the storage management environment by other factors such as data rates for a given storage device and sustainable throughput rates for a network or processor complex. The actual performance obtained by any given system will be largely dependent on these factors, but the Storage Management resource manager must be designed to handle a wide range of factors.

To meet performance expectations, the Storage Management resource manager must incorporate a wide range of techniques to ensure that data is handled in an efficient manner at the appropriate performance level. The techniques used by the Storage Management resource manager include:

- Advanced buffering (dual buffering, minimal data moves, read ahead, data caching, and memory management)
- Device blocking for efficient device use
- Data streaming for networks and storage devices
- Fast locate features for sequential devices
- Multiprocessor exploitation
- Full multitasking components

- Data striping
- Data aggregation

Scalability

The Storage Management resource manager is designed for scalability. Each system can have multiple instances of the client running simultaneously for overlapped operations. The storage management servers support multiple, concurrent operations so that the full power of the host system can be used. A single site can deploy multiple servers, all centrally managed and coordinated, to provide proper system balancing as required. Additionally, remote servers can be operated and maintained from a central location.

Multiple servers can also be connected together for hierarchical operations. With this capability, workgroup servers can provide user views into the data at any level of granularity, and the centralized servers can store the data in larger aggregates for better performance on shared, central resources such as tape robotic units. Servers are connected using the Open Blueprint Common Transport Semantics.

The Storage Management API is designed for scalability. Data objects can be large or small, and large or small numbers of these objects can be processed. Multiple instances of the API can be invoked to provide for parallel or overlapped operations as required by the application. Each API session can connect to the same or different storage management servers depending on the site's configuration.

Interoperability

Transparent file services allow different file types to exchange data when properly authorized. For example, a file that originated on a PC DOS file system could be restored to an OS/2 file system automatically.

Interoperability among different API callers is possible through coordinated data formats for the API callers. Because the Storage Management resource manager stores named binary objects, any application can restore those objects if:

- They are properly authorized to access the data
- They understand the format of the binary data

Additionally, because the Storage Management resource manager services can run on many different operating systems, the data from one system can be exported to other systems for functions such as load balancing and personnel movement.

Reliability and Availability

The Storage Management resource manager controls and protects one of an organization's most valuable resources: their data. This data must be protected against system failure, and it must be available 24 hours a day, 7 days a week.

The Storage Management resource manager incorporates many reliability and availability features to protect a customer's data. These include:

- Data replication to multiple physical storage volumes
- Off-site vaulting
- Meta data protection through mirrored databases and full backup/recovery operations
- The ability to dynamically set and update all operational values without having to shut down the system

Systems Management

The Storage Management resource manager operates effectively beneath an overall systems management umbrella. The Storage Management resource manager sends alerts to a management station for error conditions or exceeded thresholds, and the managing station sends queries to the Storage Management resource manager for regular status checks. These functions, coupled with central logging functions and a standards-based Structure Query Language (SQL) interface with Open Database Connectivity (ODBC) modules, gives the systems administrator all of the tools necessary to run and manage the Storage Management resource manager in a complex environment.

Standard protocols that are natively supported through the Storage Management resource manager's systems management agent include SNMP and Netview/MVS. New management protocols can be easily added through a modular set of frameworks in the agent.

Relationship to Other Resource Managers

The Storage Management resource manager uses and interoperates with other resource managers to varying degrees. The key resource manager that the Storage Management resource manager interoperates with is the File resource manager.

Storage management also integrates with the Open Blueprint Security resource manager to ensure protection of data and storage resources. This allows authenticated users to access the system, and allows data sharing between authorized users.

The Storage Management resource manager uses Common Transport Semantics to provide distributed management of the storage resources within an enterprise. The Directory, Transaction Manager, and HTTP resource managers are also used.

Summary

The Storage Management resource manager is a key component of IBM's Open Blueprint. It provides transparent storage management services to users of the File resource manager. It also provides for proactive management of named binary objects through its object storage API. This API allows applications to store large amounts of data without regard to communication methods, server platforms, tape device drivers, or other system-related factors.

Appendix A. Bibliography

"System-Managed Storage" by Jack P. Gelb, *IBM Systems Journal Vol 28, No 1*, 1989.

© Copyright IBM Corp. 1995, 1996

Appendix B. Notices

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. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

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:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY 10594
USA

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

AIX
AS/400
IBM
IBMLink
MVS
Open Blueprint
OS/2
RS/6000

The following terms are trademarks of other companies:

486	Intel Corporation
Intel	Intel Corporation
Novell	Novell, Incorporated
Sun	Sun Microsystems
X/Open	X/Open Company Limited

Microsoft, Windows, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

© Copyright IBM Corp. 1995, 1996

Appendix C. Communicating Your Comments to IBM

If you especially like or dislike anything about this paper, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply. Feel free to comment on specific error or omissions, accuracy, organization, subject matter, or completeness of this paper.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

- If you prefer to send comments by FAX, use this number:

  United States and Canada: 1-800-227-5088.

- If you prefer to send comments electronically, use one of these ID's:
  - Internet: **USIB2HPD@VNET.IBM.COM**
  - IBM Mail Exchange: **USIB2HPD at IBMMAIL**
  - IBMLink: **CIBMORCF at RALVM13**

Make sure to include the following in your note:

- Title of this paper
- Page number or topic to which your comment applies

© Copyright IBM Corp. 1995, 1996

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

GC23-3927-01