

DeepSpace Systems Overview

DeepSpace Documentation

Exported on 05/04/2020
DRAFT

Table of Contents

| | | |
|-----------|---|-----------|
| 1 | DeepSpace Functional Components..... | 5 |
| 2 | DeepSpace User Model | 6 |
| 3 | DeepSpace Catalog | 7 |
| 4 | Networked Systems | 8 |
| 5 | DeepSpace Server Programs | 9 |
| 6 | Related articles..... | 10 |
| 7 | Understanding How DeepSpace Works..... | 11 |
| 7.1 | The Mount Request System..... | 11 |
| 7.2 | The Catalog | 12 |
| 7.3 | Required Operation Privileges | 12 |
| 7.4 | Implementation Flow | 12 |
| 8 | DeepSpace Library Concepts | 13 |
| 8.1 | Library Maintenance..... | 13 |
| 8.2 | Library Reporting..... | 13 |
| 9 | DeepSpace Volume Management Concepts | 14 |
| 9.1 | Identifying and Naming Volumes..... | 14 |
| 9.2 | Submitting and Retrieving Volumes | 15 |
| 9.3 | Using Volumesets..... | 15 |
| 9.4 | Security | 16 |
| 9.5 | Moving Media to Different Sites | 16 |
| 9.6 | Related articles..... | 16 |
| 10 | DeepSpace Pool Concepts | 17 |
| 10.1 | Using Private Pools | 17 |
| 10.2 | Creating Private Pools..... | 17 |
| 10.3 | Related articles..... | 18 |
| 11 | DeepSpace Device Management Concepts | 19 |
| 11.1 | Accessing Devices | 19 |

| | |
|--|----|
| 11.2 Using the Device Resource Manager (DRM) | 19 |
| 11.3 Making Manual (Operator Assisted) Mount Requests..... | 20 |
| 11.4 Reclaiming Idle Devices | 20 |
| 11.5 Handling Devices and Mountpoints..... | 20 |
| 11.6 Related articles..... | 21 |

This document describes high-level concepts of DeepSpace operations, utilities, and overall functionality. Use this document to understand these concepts in preparation for a DeepSpace installation and implementation.

DeepSpace oversees nearline, removable and spin-down volume management systems. Computer users store and retrieve disk files (online files) on offline or nearline media for two main reasons:

- To make permanent, archival copies of important files.
- To free occupied and needed disk space by moving seldom-used files to nearline or offline files.

The UNIX utilities, `cpio` and `tar`, write disk files to tape, but they lack the security controls, contents catalogs, device arbitration, mount request systems, and other facilities necessary for reliable and useful nearline and offline volume activities. DeepSpace provides these services and facilities, which make removable, nearline and spin-down volume usage convenient and secure.

On this page:

- [DeepSpace Functional Components](#)(see page 5)
- [DeepSpace User Model](#)(see page 6)
- [DeepSpace Catalog](#)(see page 7)
- [Networked Systems](#)(see page 8)
- [DeepSpace Server Programs](#)(see page 9)
- [Related articles](#)(see page 10)

1 DeepSpace Functional Components

DeepSpace consists of two independent components:

- An online catalog
- A Mount Request System (MRS)

The catalog tracks what volumes exist, who owns them, and what data they contain. The MRS arbitrates competing user device requests and makes it possible for a user sitting at a terminal to ask an operator (real or virtual) to mount and unmount volumes from the library.

DeepSpace provides access to its facilities through a command-line interface (CLI) and UNIX standard I/O redirection, or through a provided C library API.

2 DeepSpace User Model

DeepSpace addresses the functional needs of three classes of users:

- System administrators
- Operators: DeepSpace places all volumes in the library under the control of the operator. The operator mounts and unmounts volumes on the system drives or mount points. When users request access to volumes, DeepSpace processes and translates each request into an instruction for the operator. In this case, it should be noted the "operator" is generally a site specific script that implements the actions required to bring the volume into a state that is ready for Input/Output.
- End users: Users submit their volumes to the library and then use them via volume access commands. If the library is set up with a public volume pool, users with proper permissions can also utilize volumes from the public pool.
- User requests are ordered in a queue and are serviced as drives become available. By controlling physical access to volumes and drives, DeepSpace maintains security over tape drives, nearline devices and user data.

3 DeepSpace Catalog

The catalog is an online database that contains the DeepSpace records. The catalog contents include:

- An entry for each volume in the library.
- An entry for each file on a volumeset.
- An entry for each pool.

DeepSpace provides several reports that extract information from the catalog.

4 Networked Systems

DeepSpace supports networks with distributed users and devices. One node on the network is designated the server and runs the DeepSpace Netmaster software. All other nodes are clients and must be equipped with the DeepSpace Netclient software. Operator interaction is the same for both stand-alone systems and network systems.



**Warning**

User IDs (UID) and Group IDs (GID) must be consistent on all nodes running DeepSpace software.

5 DeepSpace Server Programs

DeepSpace employs server programs which run in the background on each computer system. These server programs provide the coordination for mount requests, volume access, and access to the online catalog. The server node in the network runs the DS server program. It provides the catalog service and the Device Manager/Mount Request System. All nodes, including the server node, must run the DSnet server program.

6 Related articles

-  [How to Delete a Pool¹](#)
-  [How to Edit a Pool²](#)
-  [How to Create a Pool³](#)
-  [How to Edit a Volumeset Definition⁴](#)
-  [How to Delete a Volumeset⁵](#)

¹ <https://deepspace.atlassian.net/wiki/spaces/DeepSpaceDoc/pages/275480622/How+to+Delete+a+Pool>

² <https://deepspace.atlassian.net/wiki/spaces/DeepSpaceDoc/pages/275513367/How+to+Edit+a+Pool>

³ <https://deepspace.atlassian.net/wiki/spaces/DeepSpaceDoc/pages/275808257/How+to+Create+a+Pool>

⁴ <https://deepspace.atlassian.net/wiki/spaces/DeepSpaceDoc/pages/261783772/How+to+Edit+a+Volumeset+Definition>

⁵ <https://deepspace.atlassian.net/wiki/spaces/DeepSpaceDoc/pages/261488910/How+to+Delete+a+Volumeset>

7 Understanding How DeepSpace Works

DeepSpace software is based on the *client-server model* of network software. In this model, there are two primary DeepSpace software components:

- Netmaster — the server software for the master node; it includes the netclient software for the master node.
- Netclient — the client software for all computer nodes.

The main server provides services on a single node (the netmaster node) to client processes located anywhere on the network. Most DeepSpace commands can run anywhere on the network. These commands act as clients since they contact the server to perform their functions.

DeepSpace also uses *daemons*, programs that run in the background, and wait to be triggered into active processing. The DeepSpace daemon sits on all network nodes and awaits instructions from the server. This daemon starts tape reading and writing activities.

On this page:

- [The Mount Request System](#)(see page 11)
- [The Catalog](#)(see page 12)
- [Required Operation Privileges](#) (see page 12)
- [Implementation Flow](#)(see page 12)

Several programs are key to brokering DeepSpace services:

- Mount Request System (MRS)
- DeepSpace Catalog
- Device Resource Manager (DRM)

7.1 The Mount Request System

The Mount Request System (MRS) maintains a queue of mount requests and presents each in turn to the operator for action via the request monitor. The operator mounts the specified and confirms it to the request monitor. The MRS then notifies the user command which in turn reports to the user.

Is the "request monitor" the same as the Mount Request System?

Mount and unmount requests are communicated to the library operator via the request monitor; the request monitor is accessed via the dsmon program. (For more information on handling mount requests, see [Satisfying Mount Requests](#)⁶.)

⁶ <https://deepspace.atlassian.net/wiki/spaces/DeepSpaceDoc/pages/250904658/Satisfying+Mount+Requests>

7.2 The Catalog

The server maintains a catalog which tracks all tapes in the DeepSpace library and all data stored on the tapes. All access to tapes and the data on the tapes is handled via the server and the catalog.

7.3 Required Operation Privileges

The root user ID is pre-configured as an approved DeepSpace operator. To perform the operator functions, the operator must be logged in as a user with root privileges. To conduct library operations, the DeepSpace server and client software must be running on the appropriate nodes. For more information, see [How to Start, Stop, and Confirm DeepSpace Services](#)⁷.

7.4 Implementation Flow

Installing DeepSpace Software

Installation instructions are provided in the [DeepSpace Installation and Configuration Guide](#)⁸. After you install the software, and before you add any tapes to the library or perform any other configuration tasks, you may want to try the practice session outlined in [Testing a DeepSpace Implementation](#)⁹. If you are installing DeepSpace and need to change the hostname, please see [How to Change or Update a Hostname](#)¹⁰.

Adding Tapes to the Library

Before you can use DeepSpace, you must populate the DeepSpace library with storage volumes. There are two ways that this is achieved:

- The DeepSpace administrator can create a public pool and place storage volumes in it.
- Each user can submit storage volumes to the their private pool.

For more information on pools, see [Working with Pools](#)¹¹.

The submission process starts with the `dspsubmit` program. Scratch volume submission is discussed in [How to Submit a Scratch Volume](#)¹². All submitted volumes must be accepted by the library operator via the `dsaccept` program; which is discussed in [How to Add Tapes in the DeepSpace Library with the Command Line](#)¹³.

⁷ <https://deepspace.atlassian.net/wiki/spaces/DeepSpaceDoc/pages/259129422/How+to+Start%2C+Stop%2C+and+Confirm+DeepSpace+Services>

⁸ <https://deepspace.atlassian.net/wiki/spaces/DeepSpaceDoc/pages/172032073/DeepSpace+Installation+and+Configuration+Guide>

⁹ <https://deepspace.atlassian.net/wiki/spaces/DeepSpaceDoc/pages/304349187/Testing+a+DeepSpace+Implementation>

¹⁰ <https://deepspace.atlassian.net/wiki/spaces/DeepSpaceDoc/pages/232914948/How+to+Change+or+Update+a+Hostname>

¹¹ <https://deepspace.atlassian.net/wiki/spaces/DeepSpaceDoc/pages/275513345/Working+with+Pools>

¹² <https://deepspace.atlassian.net/wiki/spaces/DeepSpaceDoc/pages/275611772/How+to+Submit+a+Scratch+Volume>

¹³ <https://deepspace.atlassian.net/wiki/spaces/DeepSpaceDoc/pages/259129530/How+to+Add+Tapes+in+the+DeepSpace+Library+with+the+Command+Line>

8 DeepSpace Library Concepts

The library is the total collection of volumes and files submitted to the control of DeepSpace. Generally, only qualified operators have access to the library and its constituent volumes. The catalog tracks each volume in the library, the files on each volume as well as file and volume ownership and access permission.

On this page:

- [Library Maintenance](#)(see page 13)
- [Library Reporting](#)(see page 13)

8.1 Library Maintenance

DeepSpace tracks the number of times a volume is accessed to facilitate periodic cleaning and eventual removal. At any time, the operator can generate a list of volumes whose usage count exceeds the cleaning or removal interval. When a volume is cleaned, the operator confirms cleaning and the cleaning count is reset. When a volume is removed, the catalog entry for the volume is deleted. Volumes are only scheduled for cleaning and removal when they are in the scratch state.

8.2 Library Reporting

Both users and operators can generate a variety of reports summarizing the contents of the library. Users can generate a summary of all volumes they own as well as a detailed report on any particular volume. Operators can generate inventory reports on:

- All volumes in the library
- An individual volume
- Pools
- Volumes requiring special attention (cleaning, removal, and movement from one location to another)

For more information on DeepSpace reporting features, see [Using the DeepSpace System Reports](#)¹⁴.

¹⁴ <https://deepspace.atlassian.net/wiki/spaces/DeepSpaceDoc/pages/284688387/Using+the+DeepSpace+System+Reports>

9 DeepSpace Volume Management Concepts


Following are key concepts with regard to DeepSpace volume management.

On this page:

- [Identifying and Naming Volumes](#)(see page 14)
- [Submitting and Retrieving Volumes](#)(see page 15)
- [Using Volumesets](#)(see page 15)
- [Security](#)(see page 16)
- [Moving Media to Different Sites](#)(see page 16)
- [Related articles](#)(see page 16)

9.1 Identifying and Naming Volumes

DeepSpace uses three identifiers for each volume in the library. These identifiers are described in the following table.

| This Identifier... | Describes... |
|--------------------|---|
| Volume ID | A name (limited to 12 characters) by which users identify volumes. DeepSpace automatically generates a volume's ID unless the user supplies it. |
| Rack number | A name (limited to 12 characters) by which operators identify the physical location (or slot) of volumes; rack numbers are unique in the library. The operator must supply the rack number when the tape is accepted (dsaccept) into the library. |
| Fingerprint | <p>A coded identity derived from data on the volume; DeepSpace can initially establish a volume's fingerprint when the volume is identified with the dsid program or do so when the volume is first mounted via DeepSpace. The fingerprint is always kept current by the system. Whenever a volume is mounted, DeepSpace takes its fingerprint for comparison with the fingerprint in the catalog and corrects any mistaken mounts by the operator.</p> <div>  It is not recommended to have volumes with identical fingerprints, otherwise data may be overwritten </div> |

9.2 Submitting and Retrieving Volumes

Submitting - Submission is the process of placing volumes under DeepSpace control. Volumes are submitted to DeepSpace as a two-step process:

1. The volume owner runs the command `dspsubmit`, which creates a catalog entry for the volume and then presents the volume to an operator.
2. The operator runs the command `dsaccept` and places the volume in the library.

Retrieving - Retrieving volumes is also a two-step process:

1. Users are free to retrieve volumes from the library at any time with the command `dsretrieve`. This command confirms that the user has permission to retrieve the volume and instructs the user to approach the operator.
2. The operator runs the `dsreturn` command which tells the operator which rack number to retrieve. The operator retrieves the volume and presents it to the user.

9.3 Using Volumesets

The common unit of storage in DeepSpace is the *volumeset*.

i A volumeset consists of one or more physical volumes organized and accessed as one logically continuous volume. This volume can contain one or more data files. (Volumesets can also be submitted to the library.)


When a user writes data files to the volumeset, transitions from the end of one volume to the beginning of the next volume are handled transparently to the user. If a large file fills the first volume in the volumeset before DeepSpace has finished writing the file, DeepSpace adds another volume to the volumeset automatically. DeepSpace then instructs the operator to unmount the first volume and mount the next volume before it resumes writing the file. Volume transitions are handled similarly during requests to read data from offline volume.

DeepSpace creates volumesets by specifying the individual volumes which constitute it or by naming the *pool* / (see [DeepSpace Pool Concepts](#)(see page 17) for more information) from which its members should be drawn. Commonly, a user has a set of volumes in the user's private pool and any volumeset activity draws on these volumes for storage. A volume is added to the volumeset when the current volumes are filled and additional data is being written to the volumeset. Once a volume is made part of a volumeset, it is reserved for that volumeset until the volumeset expires.

DeepSpace supports the following standard tape labeling formats for volumesets:

- DSCM - the default.
- ANSI - ANSI formatted tapes.
- IBM - use when sharing tapes with MVS systems.
- ANSI2 - ANSI without HDR2 labels.
- IBM2 - IBM without HDR2 labels.
- TAR - a generic format usable by most utilities.
- CPIO - a generic format usable by most utilities.
- RAW - for storing raw partitions.

These formats specify how labels are written to the volumeset. DeepSpace automatically handles all issues regarding formats, so users typically can use the default format (ANSI). The other formats are available for specific circumstances (such as writing tapes that must be shared with an MVS system and storing raw partitions).

 The commands `dsvsubmit` and `dsvretrieve` operate for volumesets as `dspsubmit` and `dspretrieve` perform for individual volumes.






9.4 Security

An important aspect of a media management system is controlling access to volumes in the library. Files stored on volumesets should be as secure as files stored on disk. DeepSpace provides a UNIX-like permission mask (the `vmode` keyword of the `dsvsubmit` and `dsvedit` commands) that allows separate read/write/execute permission for owner, group, and others. To have execution permission means that you can view the catalog entry. However, only the owner can modify the volume's catalog entry. In addition to the UNIX permission mask, a volume owner can assign a password. When a volume is password protected, DeepSpace prompts for the password before allowing access to it.

9.5 Moving Media to Different Sites

For security reasons it is often wise to keep copies of important volumes at locations other than the central library. DeepSpace can track volume storage at any number of different locations. Users can request volumeset movement from one location to another with the command `dsvmove`. Volumes are actually moved by the operator who periodically generates a list of all volumes waiting for movement and transports them to their new location. The administrator controls what sites are available and whether to accept mount requests for volumes stored at each site.

9.6 Related articles

-  [How to Edit a Volumeset Definition¹⁵](#)
-  [How to Delete a Volumeset¹⁶](#)
-  [How to Create a Volumeset from Pool Volumes¹⁷](#)
-  [How to Add a Volumeset into the Library¹⁸](#)
-  [How to Submit a Scratch Volume¹⁹](#)

¹⁵ <https://deepspace.atlassian.net/wiki/spaces/DeepSpaceDoc/pages/261783772/How+to+Edit+a+Volumeset+Definition>

¹⁶ <https://deepspace.atlassian.net/wiki/spaces/DeepSpaceDoc/pages/261488910/How+to+Delete+a+Volumeset>

¹⁷ <https://deepspace.atlassian.net/wiki/spaces/DeepSpaceDoc/pages/261587131/How+to+Create+a+Volumeset+from+Pool+Volumes>

¹⁸ <https://deepspace.atlassian.net/wiki/spaces/DeepSpaceDoc/pages/261554379/How+to+Add+a+Volumeset+into+the+Library>

¹⁹ <https://deepspace.atlassian.net/wiki/spaces/DeepSpaceDoc/pages/275611772/How+to+Submit+a+Scratch+Volume>

10 DeepSpace Pool Concepts

A pool is a collection of volumes. A volume can belong to only one pool. Each volume in a pool is in one of two states:

- Scratch - A scratch volume is not in current use and does not contain any data. For example, when a user submits a blank volume to the library it enters the catalog as a scratch volume in the user's private pool.
- Active - An active volume, in contrast, already has data stored on it.

On this page:

- [Using Private Pools](#)(see page 17)
- [Creating Private Pools](#)(see page 17)
- [Related articles](#)(see page 18)

10.1 Using Private Pools

Each user has a pool named *private* which, by default, contains all of the user volumes not explicitly assigned to another pool. The *private pool* is not created until it is explicitly referenced by name through any of the DeepSpace commands.

10.2 Creating Private Pools

Private pools can be created via the `dss` command: for example:

```
dss pool=private pinfo
```

A user can create pools as needed and move volumes from one pool to another if they own that pool. However, a user cannot submit or remove volumes from a pool unless they own it.

i The full specification of pool is `userid/pool`. If `userid` is omitted, the current effective user ID is assumed. Only operators may create pools for IDs other than their own.

The user can also specify what other users may access particular volumes within a pool. For example, to create a public pool a user would enter the following:

```
dsscreate uacc=ANY gacc=ANY dslib/oz
dsssubmit type=9840 pool=dslib/oz
```

To access the public pool, the user would then enter the following:






```
dsscreate pool=dslib/public type=9840 toto
```

To access a public pool with no current members you must have the automatic volume addition flag set to yes. To do this enter:

```
dspcreate uacc=ANY gacc=ANY padd=yes dslib/oz
```

When a user creates a volumeset, the user defines the pool(s) from which DeepSpace should draft volumes into service for the volumeset. As needed, DeepSpace acquires a volume from the designated pool and makes it part of the volumeset. The volume goes from scratch to active status while it is a member of the volumeset. When the volumeset is deleted, the volume returns to scratch status and is available for further use.

10.3 Related articles

-  [How to Delete a Pool](#)²⁰
-  [How to Edit a Pool](#)²¹
-  [How to Create a Pool](#)²²
-  [DeepSpace Systems Overview](#)(see page 4)
-  [DeepSpace Pool Concepts](#)(see page 17)

²⁰ <https://deepspace.atlassian.net/wiki/spaces/DeepSpaceDoc/pages/275480622/How+to+Delete+a+Pool>

²¹ <https://deepspace.atlassian.net/wiki/spaces/DeepSpaceDoc/pages/275513367/How+to+Edit+a+Pool>

²² <https://deepspace.atlassian.net/wiki/spaces/DeepSpaceDoc/pages/275808257/How+to+Create+a+Pool>

11 DeepSpace Device Management Concepts

DeepSpace is network software. It provides a universal volume management service to multiple systems on a network. All drives and mountpoints are centrally controlled and are accessible for volume operations. A user can access volumes via devices on another network node just as if it were on a local node. See the following for more information on how devices are managed in a DeepSpace environment.

On this page:

- [Accessing Devices](#)(see page 19)
- [Using the Device Resource Manager \(DRM\)](#)(see page 19)
- [Making Manual \(Operator Assisted\) Mount Requests](#)(see page 20)
- [Reclaiming Idle Devices](#)(see page 20)
- [Handling Devices and Mountpoints](#)(see page 20)
- [Related articles](#)(see page 21)

11.1 Accessing Devices

Two components can be used to manage devices:

- The Device Resource Manager (DRM) - provides a device reservation facility.
- The Mount Request System (MRS) - coordinates operator assisted volume mounts.

Combined, these two components provide users with a method for interacting with online, mounted volumes on an ad hoc basis. A user only needs to specify the volume or volumeset to be mounted. DeepSpace automatically determines the type of drive needed, reserves it, instructs the operator to mount the volume, and then electronically verifies that the correct volume indeed has been mounted.

11.2 Using the Device Resource Manager (DRM)

The DRM maintains a prioritized queue of users waiting to reserve tape drives and mountpoints. When either becomes available, the DRM assigns the device to the next request in line. To avoid deadlocks, the DRM requires users who need simultaneous access to more than one device to reserve all necessary devices with a single reservation request.

The command `dsreserve` reserves devices. An example of a `dsreserve` command is shown below. In the example, two tape drives are reserved: one capable of supporting 9840 and one capable of supporting 9940.

```
dsreserve type=9840,9940 as D1,D2
```

Once you have completed your tape activity, you should free devices with the command `dsfree`. The example below frees the drives reserved by the `dsreserve` example:

```
dsfree D1
dsfree D2
```

11.3 Making Manual (Operator Assisted) Mount Requests

Users access volumesets via the `dsvaccess` command. This command automatically reserves an appropriate device for the specified volumeset (if one is not already reserved). For example, the following command requests the volumeset named `testset`:

```
dsvaccess testset
```

After the volumeset is accessed, the user can read and write files with the `dsvread` and `dsvwrite` commands. An example of `dsvread` is shown below. In the example, the first three files off the volumeset accessed in the previous example are read into files named `tf1`, `tf2`, and `tf3`.

```
dsvread > tf1
dsvread > tf2
dsvread > tf3
```

The `dsvread` example below demonstrates how the command can be used to write only the third file on the volumeset to a file:

```
dsvread fseq=3 > tf3
```

Volumeset access is terminated with the `dsvrelease` command. The example shown below releases the volumeset that was accessed in the previous examples.

```
dsvrelease
```

The `dsvaccess` and `dsvread` commands can handle IBM, ANSI, TAR, and CPIO volumesets. A RAW volume type is available for tapes that do not fit one of these classifications.

DeepSpace handles multi-volume volumesets transparently to the user. The library operator is instructed to mount and unmount the volumes in the volumeset as needed to satisfy read and write requests that span volumes.

11.4 Reclaiming Idle Devices

Occasionally users access volumesets and forget about them. As a result, devices are needlessly tied up. DeepSpace has a facility to reclaim devices that accumulate excessive idle time. In the event a user logs out without freeing a device, DeepSpace reclaims the device after a time-out and makes it available to other users. The default time-out is 20 minutes.

11.5 Handling Devices and Mountpoints

DeepSpace automatically assigns devices and mountpoints to users via their requests for access to volumesets. Each access request to a volumeset causes DeepSpace to assign an appropriate device for the user's session. When there are more requests than there are devices, then the requests are queued and satisfied as devices become available. The online catalog stores information about library volumes, volumesets, files, pools, and rotations. Several user commands are available to perform catalog activities. For more information, see [Using DeepSpace Storage Management](https://deepspace.atlassian.net/wiki/spaces/DeepSpaceDoc/pages/261488763/Using+DeepSpace+Storage+Management)²³.

²³ <https://deepspace.atlassian.net/wiki/spaces/DeepSpaceDoc/pages/261488763/Using+DeepSpace+Storage+Management>

11.6 Related articles

 [Working with Tape Devices](#)²⁴

²⁴ <https://deepspace.atlassian.net/wiki/spaces/DeepSpaceDoc/pages/278986776/Working+with+Tape+Devices>