Chapter 18
-
JFS/OnlineJFS

INDEX

Basics of JFS 3.3

JFS is an extent-based journaling file system. Its journaling, or intent-logging, capability provides accelerated recovery after system failure, and increases performance for NFS and DBMS applications. The File system storage layout uses contiguously allocated sets of blocks, or extents. This layout improves performance by optimizing physical disk I/O to an application's needs, in addition to reducing the number of index structures required to access file data. It also incorporates on-line utilities that allow common administrative tasks (including stable backup, defragmentation, and resizing) to be performed without reducing data availability.

JFS 3.3 is only supported at HP-UX 11.X

New features of JFS 3.3

(the features in need of explanation will be discussed below)

- Fast file system recovery
- No-panic error limits on disk and file system failures
- Extent-based allocation of disk space to files
- Quota capabilities for controlling strategic resources
- Supported across multi-vendor systems and disks
- new packaging and runtime licensing strategy for OnLineJFS
- new disk layout, version 4
- support for Access Control Lists (ACLs)
- new vxtunefs command for modifying file system parameters
- performance enhancements
- file system shrink enhancement

Supported file and file system sizes

Although it may be possible to create files or file systems larger than these documented limits, such files and file systems are not supported and the results of using them may be unpredictable.

JFS (VxFS) supported sizes

HP-UX Release	HP JFS Version	Veritas Disk Layout Version	Maximum File Size	Maximum File System Size
UX 10.01	JFS 2.0	**Version 2**	2 GB	4 GB
UX 10.10	JFS 2.0	**Version 2**	2 GB	128 GB
UX 10.20	JFS 3.0	Version 2	2 GB	128 GB

		Version 3	128GB	128 GB
UX 11.00	JFS 3.1	Version 2	2 GB	128 GB
		Version 3	1 TB	1 TB
	JFS 3.3	Version 2	2 GB	128 GB
		Version 3	1 TB	1 TB
		Version 4	1 TB	1 TB
UX 11.11	JFS 3.3	Version 2	2 GB	128 GB
		Version 3	2 TB	2 TB
		Version 4	2 TB	2 TB

bold default disk layouts for particular HP-UX Release/JFS version.

NOTE:
for UX 11.00 with disk layout version 3 PHKL_22719 (or newer) is needed to avoid mount problems if extending or creating file systems beyond 128 GB.

and just to be complete

HFS supported sizes

HP-UX Release	Maximum File Size	Maximum File System Size
UX 10.01	2 GB	4 GB
UX 10.10	2 GB	128 GB
UX 10.20	128 GB	128 GB
UX 11.00	128 GB	128 GB
UX 11.11	128 GB	128 GB

NOTE: As of UX 10.20 it is possible to exceed the 128 GB limit to 256 GB, but it is not supported.

Supported disk layouts and features

Three disk layouts are available with the VxFS file system:

Version 2

The Version 2 disk layout was designed to support features such as filesets, dynamic inode allocation, and enhanced security.

Version 3 (default for 10:20,11.00)

The Version 3 disk layout encompasses all file system structural information in files, rather than at fixed locations on disk, allowing for greater scalability. Version 3 supports files and file systems up to one terabyte in size.

Version 4 (default for 11.11)

The Version 4 disk layout supports access control lists (ACLs). Further it has an enhanced ability to shrink a file system by moving extents off the area of the file system being reduced. However, if JFS cannot move extents off the area of the file system being reduced, the shrink will fail.

Upgrading JFS 3.1 file systems to JFS 3.3 (disk layout version 4)

The upgrade of a file system from an older disk layout to version 4 can be made by using the command vxupgrade. The /root and /usr file systems cannot be upgraded to disklayout 4. Further /var and /opt should not be upgraded because these both file systems are core file systems which are crucial for system recovery. A file system with disklayout 2 cannot be directly upgraded to disklayou 4, first it must be upgraded to disklayout 3.

1) Display the current disk layout:

```
# /sbin/fs/vxfs3.3/vxupgrade /mountpoint
```

2) Upgrade the file system:

```
# /usr/sbin/vxupgrade –n <m> -r /dev/vgXX/rlvolX /mountpoint
```

`-n` <m> is new version (4 in this case)
`-r` raw device file

NOTE: an upgraded file system cannot be downgraded.

Access Control Lists - ACLs

ACL is a new tool which gives a more precise way to control access to file than with traditional Unix file permissions. An ACL contains one-line entries naming specific users and groups and indicating what access is granted to each. The presence of an ACL also changes the meaning of group permission bits displayed using the `ls –l` command. An ACL has at least four entries: a user entry, a group entry a class entry and an other entry. This is called a minimal ACL, which represents the same permissions as the standard UNIX system permission bits. To grant or deny access to specific users and groups on this system, there is the possibility to add up to 13 more user and group entries to the four minimal entries. In this case the `chmod` affects only the class ACL entry.: Note: This new feature is only supported by the new disk layout version 4 (the default disk layout for HPUX 11.0 is 3)

Quotas

Quotas serves to limit the file system and datablock usage for specific users. There are two limits, which must be set, the soft and the hard limit. The soft limit can be exceeded for a given time limit. This time limit is per default 7 days and can be changed via `edquota –t`. When a user exceeds the soft limit a warning is displayed. The hard limit is an absolut limit, which cannot be exceeded, so the soft limit should be lower than the hard limit.
To use quotas a file quotas must exist in the root directory of the file system. The quotas are set by using edquota which opens an editor to set the soft and hard limits. The quotas can be turned on either by the command quotaon or by the mountoptions (`-o quota`)

vxtunefs

This command can be used to print or set tunable I/O parameters of mounted file systems. The I/O parameters are:

- Parameters describing the I/O properties of the underlying device
- Parameters to indicate when to treat an I/O as direct I/O
- Parameters to control the extend allocation policy for the specific file systems

It works on a list of mountpoints specified on the commandline or all the mounted file systems listed in the tunefstab file, which is in `/etc/vx/tunefstab`. The change of the parameters takes place immediately.

Basics of OnlineJFS

Online JFS is an optional product, that needs to be purchased seperately. It adds to Base JFS all online administration features. Those features are very usefull for HA configurations, as they allow non-disruptive systemadministration, and it is much recommended that customers wanting a highly available system purchase Online JFS.

Features of OnlineJFS

- Online file system resizing
- Online snapshot with support for common backup utilities
- Online file defragmentation and directory optimization
- Discovered direct I/O for optimal through-put of large reads/writes
- Support for standard DMAPI interfaces as defined by X/Open
- High integration with other VERITAS storage management applications
- Enablers for non-HP VERITAS Edition functionalities such as Quick I/O
- Manipulation of extent attributes via setext

What is a Snapshot?

A snapshot is a method of an online backup of data. An image of a mounted file system becomes an exact read-only copy of the file system at a certain point in time. This feature offers the possibility to backup either selected files via `cpio` or the entire file system via `vxdump` etc. .
The snapshot volume should be about 10-20% the size of the original file system. The snapshot volume need not be structured in any way it is created by mountoptions. It is not necessary to execute `newfs` for a snapshot file system.

NOTE: a snapped file system can´t be umounted before all snapshots are umounted. Once a snapshot is umounted it is lost.

To display the features of the snapshot the command `snapbdf` can be used. The `snapbdf` command, developed by the Expert Center team, reports amount of used and free space on all snapshot file systems it finds active on a system. Reported numbers are in kilobytes. Currently it supports UX 10.20, 11.00 (both JFS 3.1 and 3.3) and HP-UX 11.11.

Example:

```
# ./snapbdf
Filesystem             kbytes     used    avail  %used Snap of
/dev/vg00/lvol13        40960      841    40119     2% /dev/vg00/lvol6

# bdf /tmp /mnt/snapshot
Filesystem          kbytes    used   avail %used Mounted on
/dev/vg00/lvol6     204800  170255   33744   83% /tmp
/dev/vg00/lvol13    204800  170255   32386   84% /mnt/snapshot
```

snapbdf can be downloaded at:
http://wtec.cup.hp.com/~hpux/fs/VxFS/tools.htm (HP internal)

Using extent attributes

Online JFS 3.3 provides the possibility to specifies a fixed extent size for a file and reserve space for a file using the command setext. The file must already exist. Other policies determine the way these attributes are expressed during the allocation process. The user can specify that:

- the space reserved for a file must be contiguous
- no allocations should be made for a file beyond the current reservation
- unused reservation should be released when the file is closed
- space should be allocated but no reservation should be assigned
- the file size should be changed to immediately incorporate the allocated space

Fixed extent sizes or alignment will cause the file system to return an error message reporting insufficient space if no suitably sized (or aligned) extent is available. This may happen even if the file system has plenty of free space and the fixed extent size is large. To search for the files which are affected the command ls –le can be used

```
# ls –le /tmp
-rw-r--r--   1 root     sys       266 Oct  2 16:22 temp_588
-rw-rw-rw-   1 root     sys         0 Feb  6 06:47 test  :res 2048 ext 2048
```

NOTE: to free the allocated space, the affected files must be removed.

Troubleshooting File System Problems

There are three different ways how a file system reacts to problems:

- marks an inode bad
  this happens if an inode update or a directory- block update fails. In this case any attempt to access the data in the file or change the inode will fail.
- disables transactions
  this happens if the file system detects an error while writing to the intent log. In this

case block or inode frees or allocation, structural changes, directory entry changes will fail.

- disables the file system
  this happens if a superblock update fails.

Anytime the file system encounters a problem it will write an error message to the `syslog.log` (all error messages are mentioned in the JFS 3.3 administration guide). In nearly every case the file system must be umounted and a `fsck` has to be performed. Main reason for file system errors result from hardware failures, hence it is recommended to check the syslog for I/O errors.

Installation of JFS 3.3 and OnlineJFS

Installation

Contrary to the older Versions - Base JFS and Online JFS 3.3 are in one binary file instead of separate files as in previous releases. Base JFS is a code-restricted subset of the OnlineJFS product. In order to use OnlineJFS, customers obtain a license key from Electronic Software and Information (ESI), then use the license key to activate OnlineJFS. To get the license key a request form which can be found at:
http://www3.grenoble.hp.com/hermes/Licensing/Forms/cdw9000o.toc.html (HP internal)
must be sent to the following address:

Hewlett-Packard
Software & Information Delivery Europe (SIDE)
5, avenue R. Chanas
38053 Grenoble Cedex 9
France

Phone: +33.(0)4.76.14.15.29
Fax: +33.(0)4.76.14.25.15
Email: codeword_europe@hp-france-gen1.om.hp.com
Hours: monday through friday, 8am-5pm GMT+1

On HP-UX 11.0, JFS 3.3/OnLineJFS 3.3 requires a patch bundle to be installed prior to installing JFS 3.3/OnLineJFS 3.3. The "B3929CA-foundation" patch bundle is included with the B3929CA product on the Independent Product CD.
To do the installation follow these steps:

1) Download the JFS 3.3 depot file, B8861E_11_00.depot

2) Move the file to `/tmp`

3) Verify that the bundle has downloaded correctly using `swlist`:

```
# swlist -d @ /tmp/B8861E_11_00.depot
#
# Initializing...
# Contacting target "localhost"...
# Target: localhost:/tmp/B8861E_11_00.depot
# Bundle(s):

B3929CA B.03.03 JFS 3.3 file system For 11.00
B3929CA-foundation B.11.00 Foundation Patch Bundle for JFS 3.3
```

4) Check that you are working on an HP-UX 11.00 system:

```
# uname -r
B.11.00
```

If the result of the uname command is anything but B.11.00, do not continue.

5) Install the Foundation Patch Bundle for JFS 3.3:

```
# swinstall -x autoreboot=true
            -s /tmp/B8861E_11_00.depot B3929CA-fndation
```

Your system will reboot.

6) After your system reboots, install the JFS 3.3 software:

```
# swinstall -x autoreboot=true -s /tmp/B8861E_11_00.depot JFS
```

Your system will reboot.

7) To enable OnlineJFS all you have to do is the following:

```
# swinstall -x ask=true -s /var/tmp/B3929CA.depot OnlineJFS
```

Your system will NOT reboot.

Enter License Key

```
# /sbin/fs/vxfs3.3/vxenablef –a
```

Uninstallation

The removal must be done in the reverse direction. To uninstall OnlineJFS and get back to JFS, swremove OnlineJFS. swremove executes the remove script that removes the vxlicense key installed. By removing the vxlicense key, OnlineJFS will be disabled. After that JFS 3.3 can be removed and JFS 3.1 will be enabled. (It is not necessary to remove the patches). If JFS 3.3 is removed there is no possibility to use any file systems that have been upgraded to disk layout version 4. So before JFS 3.3 is removed the files must be downgraded using the following procedure:

1) Back up JFS 3.3 file system data.
2) Unmount the file systems.
3) Edit `/etc/fstab` to prevent mounting the JFS 3.3 file systems at system startup.
4) Remove JFS 3.3 with the Software Distributor (SD-UX) `swremove` utility.
5) Recreate the JFS file systems on the appropriate devices (logical volumes) using `newfs(1M)`; they will be created as JFS 3.1 file systems.
6) Mount the file systems using the appropriate mount-point directories.
7) Restore the saved data to the file systems.

8) Edit `/etc/fstab` to allow the restored JFS 3.1 file systems to be mounted at system startup.

Similar features of JFS and OnlineJFS

Creation of a JFS file system

JFS/OnlineJFS

`# newfs -F vxfs /dev/vgXX/rlvolX` ***(for the default disk layout 3)***

-F = file system type

Creation of a file system with largefile-option (files larger than 2GB):

`# mksf –F vxfs –o version=4 /dev/vgXX/rlvolX` ***(for disk layout 4)***

Mounting a JFS file system

JFS/OnlineJFS

```
# mount <option> <specific option> /dev/vgXX/lvolX <mountpoint>
```

You can choose between the following `<options>`:

`-F` = file system type
`-a` = mounts all file systems which are mentioned in /etc/fstab
`-e` = displays all file systems which are mounted
`-l` = limit actions to local file systems only
`-o` = to set specific options

`<specific option>:`

a list of comma-separated suboptions and/or keyword/attribut pairs. There are the following specific options:

1) Miscellaneos <specific options>

`rw`	read-write (**default**)
`suid`	set-user ID execution allowed (**default**)
`ro`	read only
`nosuid`	set-user-ID execution not allowed
`quota`	disk quotas enabled
`remount`	change of mountoptions for a mounted file system

2) Intent Log <specific_options>

`log`	all metadata changes are logged immediately (**default**)
`delaylog`	only critical metadata changes are logged immediately
`tmplog`	metadata changes logging is almost always delayed
`nolog`	intend log is disabled

3) Write <specific_options>

`direct`	direct writes
`dsync`	datasynchronous writes
`closesync`	sync on close writes
`tmpcache`	temporary caching

Further there is the possibility to determine how ordinary/synchronous writes are treated (only for Online JFS):

`mincache = direct|dsync|closesync|tmpcache`	ordinary writes
`convosync = direct|dsync|closesync|tmpcache`	synchronous writes

4) Other <specific_options>

`blkclear`	all data extents are cleared before being allocated to a files

Displaying features of the file system

JFS/OnlineJFS

There are two commands available to display the file system features (for example to check the disk layout or if the file system supports large files).

```
# mkfs –m /dev/vgXX/lvolY
```

displays the commandline which created the file system

```
# mkfs -m /dev/vg00/lvol5
# mkfs -F vxfs -o
ninode=unlimited,bsize=1024,version=4,inosize=256,logsize=1024,largefiles
/dev/vg00/lvol5 20480
```

```
# fstyp –v /dev/vgXX/rlvolY
```

displays the contents of the superblock

```
# fstyp -v /dev/vg00/rlvol5
vxfs
version: 4
f_bsize: 8192
f_bsize: 8192
f_frsize: 1024
f_blocks: 20480
f_bfree: 18625
f_bavail: 17717
f_files: 3920
f_ffree: 4656
f_favail: 4656
f_fsid: 1073741829
f_basetype: vxfs
f_namemax: 254
f_magic: a501fcf5
f_featurebits: 0
f_flag: 16    <= Largefiles
f_fsindex: 5
f_size: 20480
```

Repair of an inconsistent file system

JFS/OnlineJFS

```
# umount <mountpoint>

# fsck –F vxfs [-o full] [-y] /dev/vgXX/rlvolX
```

operation detects damage, an indication that a complete check is required is placed in the super-block, and a full check is performed.

or if the intent-log-area is damaged:

```
# fsck –F vxfs –o full,nolog [–y] /dev/vgXX/rlvolX
```

The „nolog“ option repair without logrepair. This option may be used if the log area was physically damaged.

```
# mount –F vxfs /dev/vgXX/lvolX <mountpoint>
```

Tuning of I/O parameters

JFS/OnlineJFS

1) print the tuning parameters for all file systems specified on the command line:

```
# vxtunefs –p [{<mountpoint>|<block-special>}]
```

2) setting of new parameters for the file system:

```
# vxtunefs –s –o <parameter=value> [{<mountpoint>|<block-special>}]
```

`-s` = sets the parameter

possible parameters:

(please refer to the manual page for detailed description):

`read_pref_io`	preferred read request size. Default value is 64K
`read_nstream`	number of parallel read requests of size `read_pref_io`. Default value is 1
`write_pref_io`	preferred write request size Default value is 64K
`write_nstream`	number of parallel write requests of size `write_pref_io`. Default value is 1
`default_indir_size`	increases default indirect extent size
`discovered-direct_ioz`	gives the value of file I/O request thenceforward the I/O request is handled as discovered direct I/O
`initial_extent_size`	changes the default initial extent size

`max_buf_data_size`	maximum buffer size allocated for file data (8K or 64KB) default is 8K
`max_direct_iosz`	maximum size of a direct I/O request that will be issued by the file system
`max_diskq`	limits the maximum disk queue generated by single file. Default is 1MB
`max_seqio_extent_size`	increases or decreases the maximum size of extent

Quotas

JFS/OnlineJFS

Creating the quotas file under the root directory of the file system

```
# touch /<moutpoint>/quotas (must be owned by root)
```

setting quotalimits for a via edquota (can be used like `vi`)user:

```
#edquota <username>
e.g.
fs /home blocks (soft = 10, hard = 20) inodes (soft = 10, hard = 20)
fs / blocks (soft = 10, hard = 20) inodes (soft = 10, hard = 20)
```

edit the limits

```
#quotaon

#quotacheck <mountpoint>
```

#quota –v <username> to display the quotas

Example:

```
# quota -v darrit
Disk quotas for darrit (uid 103):
File system     usage  quota  limit    timeleft  files  quota  limit    timeleft
/home               7     10     20           7     10     20
```

The quotas can also be turned on by the mountoptions

```
#mount –F vxfs –o quota /dev/vgXX/lvolY/ /mountpoint
```

To turn off quota use

```
# quotaoff
```

Acess Control Lists (ACL)

JFS/OnlineJFS

Note: only available for disk layout 4

1) Display an access control list (ACL) for files (JFS File systems only)

```
# getacl <file>

# file: filename
           owner: uid
           group: gid
           user::perm
           user:uid:perm
           group::perm
           group:gid:perm
           class:perm
           other:perm
           default:user::perm
           default:user:uid:perm
           default:group::perm
           default:group:gid:perm
           default:class:perm
           default:other:perm
```

2) setacl - modify access control lists (ACLs) for files (JFS File systems only)

Example:
To add one ACL entry to file “testfile”, giving user root read permission only, type:

```
# setacl -m user:root:r - - testfile


# getacl testfile

# file: testfile
# owner: root
# group: sys
user::rw-
user:root:r--
group::r--
class:r--
other:r–
```

You have also the possiblity to define the ACL List within a testfile.acl to convert the entries to the testfile.

For further specific options refer to the manual page

Erasing a file system

JFS/OnlineJFS

Remember that all data on an erased file system will be lost.
`# umount <mountpoint>`
`# lvremove /dev/vgXX/lvolX`
or `newfs/mkfs` on the concerning logical volume

Features that are Treated Differently between JFS and OnlineJFS

Extension of a file system

JFS	Online JFS

`# umount <mountpoint>`
`# lvextend –L <n> /dev/vgXX/lvolX`
`-L <n>` set new size of lvol to `n` MB
`# extendfs –F vxfs /dev/vgXX/rlvolX`
`# mount <mountpoint>`
	`# lvextend –L <n> /dev/vgXX/lvolX`
	`-L <n>` set new size of lvol to `n` MB
	`# fsadm –F vxfs –b <n> <mountpoint>`
	`-b <n>` set new file system size to `n` **KB**

Reduction of a file system

JFS	Online JFS

Before reduction a backup of the data must be done otherwise the data will be lost
`# umount <mountpoint>`
`# lvreduce -L <n> /dev/vgXX/lvolX`
`-L <n>` set new size of lvol to `n` MB
`# newfs –F vxfs /dev/vgXX/rlvolX`
`# mount /dev/vgXX/lvolX <mountpoint>`
	Only file systems which are multiple of 32 MB can be reduced online (it depends on the allocated extent distribution). A backup before execution is recommended
	`# fsadm –F vxfs –d –D –e –E <mountpoint>`
	`-d` perform directory defragmentation
	`-e` perform extent defragmentation
	`# fsadm –F vxfs –b <n> <mountpoint>`
	`-b <n>` set new file system size to `n` **KB**
	`# lvreduce –L <n> /dev/vgXX/lvolX`
	`-L <n>` set new size of lvol to `n` MB

Enabling large file support

JFS	Online JFS

`# umount <mountpoint>`
`# fsadm -F vxfs -o largefiles /dev/vgXX/rlvolX`
`# mount -F vxfs -o largefiles /dev/vgXX/lvolX <mountpoint>`
entry in /etc/fstab (example):
`/dev/vgXX/lvolX <mountpoint> vxfs delaylog,largefiles 0 2`
check the largefile option:
`# fsadm -F vxfs <mountpoint>`
	`# fsadm -F vxfs -o largefiles <mountpoint>`
	entry in /etc/fstab (example):
	`/dev/vgXX/lvolX <mountpoint> vxfs delaylog,largefiles 0 2`
	check the largefile option:
	`# fsadm -F vxfs <mountpoint>`

Features Only Available for OnlineJFS

Defragmentation of a file system

OnlineJFS

```
# fsadm –F vxfs –d –D –e –E <mountpoint>
```

`-d`	performs a directory defragmention
`-D`	reports on directory fragtmention
`-e`	performs a file extent defragmentation
`-E`	reports on file extent fragmation

Using extent attributes

OnlineJFS

Extent attributes can be manipulate using setext

```
#setext –e <extentsize> -r <reservationsize> -f <flag> <file>
```

`possible flags:`

`align:` Specify that all extents must be aligned on extent_size boundaries relative to the start of allocation units

`chgsize:` Immediately incorporate the reservation into the file and update the file's on-disk inode with size and block count information that is increased to include the reserved space.The space added to the file is not initialized

`contig:` Specify that the reservation must be allocated contiguously

`noextend:` Specify that the file may not be extended after the preallocated space is used

`noreserve:` Specify that the reservation is not a persistent attribute of the file. Instead, the space is allocated until the final close of the file, when any space not used by the file is freed. The temporary reservation is not visible to the user

`trim:` Specify that the reservation is reduced to the current file size after the last close by all processes that have the file open

The extend Attributes can either be seen by the commands

```
#getext <filename>
```

or

```
#ls -le
```

Creation of a snapshot

OnlineJFS

1) Creation of a snapshot LV and file systems (should be 10-20% of the snapped file system):

```
# lvcreate –L 80 –n lvsnap vgXX
```

`-L` size of lvol
`-n` name of lvol

```
# mount –F vxfs –o snapof=/org_vol /dev/lvsnap /aux_dir
```

`/org_vol` = file system, which has to backed up
`/aux_dir` = auxiliary directory for mounting snapshot

2) Backup from snapshot file system

```
# cd /aux_dir
# tar cvf /dev/rmt/0m
```

or

```
# find . | cpio –ocvx >/dev/rmt/0m
```

3) Umount the snapshot

```
# umount /aux_dir
```

Additional information

WTEC JFS/VxFS Web Site:

http://wtec.cup.hp.com/~hpux/fs/jfs_vxfs.htm (HP internal)

Manual Pages

`newfs(1M)`, `extendfs(1M)`, `mount(1M)`, `fsadm(1M)`, `getext`
(there are man pages for vxfs and hfs, e.g. `newfs_vxfs(1M)` or `mount_hfs(1M)`)

New man pages with JFS 3.3:
`getacl(1)`, `setacl(1)`, `vxenablef(1M)`, `vxlicense(1M)`, `vxtunefs(1M)`, `acl(2)`, `aclsort(3C)`, `aclv(5)`, `setext`

Technical Documentation

- HP JFS 3.3 and HP OnLineJFS 3.3 VERITAS File System 3.3 System Administrator's Guide
- HP JFS 3.3 and HP OnLineJFS 3.3 Release Notes
- HP JFS 3.3 Access Control Lists

can be found at

http://docs.hp.com/hpux/10.x/index.html#System%20Administration	(UX 10.X)
http://docs.hp.com/hpux/11.0/index.html#System%20Administration	(UX 11.00)
http://docs.hp.com/hpux/11i/index.html#System%20Administration	(UX 11.11)

Trainings

A Level 200 Self-paced training for JFS/OnlineJFS 3.3 is being offered at
http://cso.fc.hp.com/ssil/uxsk/ (HP internal)