NAME

mount_cachefs – mount CacheFS file systems

SYNOPSIS

mount –**F cachefs** [*generic_options*]
 –**o backfstype=***file_system_type* [*other cacheFS options*] *special mount_point*

DESCRIPTION

The CacheFS-specific version of the **mount** command mounts a cached file system; if necessary, it NFS-mounts its back file system. It also provides a number of CacheFS-specific options for controlling the caching process. For more information regarding back file systems, refer to the *System Administration Guide, Volume I.*

OPTIONS

To mount a CacheFS file system, use the generic **mount** command with the –**F** option followed by the argument **cachefs**. The following generic **mount** options are available:

–**m** Mount the file system without making an entry in the **/etc/mnttab** file.

–**O** Overlay mount. Allows the filesystem to be mounted over an existing mount point, making the underlying filesystem inaccessible. If a mount is attempted on a pre-existing mount point without setting this flag, mount will fail with the error:
 mount –**F cachefs: mount failed Device busy**.

–**r** Mount the file system read-only.

The following arguments to the –**o** option are specifically for CacheFS mounts. Use commas to separate multiple options. Note: the **backfstype** argument must be specified.

backfstype=*file_system_type*
 The file system type of the back file system (for example, **nfs**).

backpath=*path* Specifies where the back file system is already mounted. If this argument is not supplied, CacheFS determines a mount point for the back file system. The back file system must be read-only.

cachedir=*directory*
 The name of the cache directory.

cacheid=*ID* *ID* is a string specifying a particular instance of a cache. If you do not specify a cache ID, CacheFS will construct one.

write-around | **non-shared**
 Write modes for CacheFS. The **write-around** mode (the default) handles writes the same as NFS does; that is, writes are made to the back file system, and the affected file is purged from the cache. You can use the **non-shared** mode when you are sure that no one else will be writing to the cached file system. In this mode, all writes are made to both the front and the back file system, and the file remains in the cache.

noconst Disables cache consistency checking. By default, periodic consistency checking is enabled. Specify **noconst** only when you know that the back file system will not be modified. Trying to perform cache consistency check using **cfsadmin** –**s** will result in error. **demandconst** and **noconst** are mutually exclusive.

demandconst Verifies cache consistency only when explicitly requested, rather than the periodic checking that is done by default. A consistency check is requested by using the –**s** option of the **cfsadmin**(1M) command. This option is useful for back file systems that change infrequently, for example, **/usr/openwin**. **demandconst** and **noconst** are mutually exclusive.

local-access Causes the front file system to interpret the mode bits used for access checking instead or having the back file system verify access permissions. Do not use this argument with secure NFS.

purge Purge any cached information for the specified file system.

rw | **ro** Read-write (default) or read-only.

suid | **nosuid** Allow (default) or disallow set-uid execution.

acregmin=*n* Specifies that cached attributes are held for at least *n* seconds after file modification. After *n* seconds, CacheFS checks to see if the file modification time on the back file system has changed. If it has, all information about the file is purged from the cache and new data is retrieved from the back file system. The default value is **30** seconds.

acregmax=*n* Specifies that cached attributes are held for no more than *n* seconds after file modification. After *n* seconds, all file information is purged from the cache. The default value is **30** seconds.

acdirmin=*n* Specifies that cached attributes are held for at least *n* seconds after directory update. After *n* seconds, CacheFS checks to see if the directory modification time on the back file system has changed. If it has, all information about the directory is purged from the cache and new data is retrieved from the back file system. The default value is **30** seconds.

acdirmax=*n* Specifies that cached attributes are held for no more than *n* seconds after directory update. After *n* seconds, all directory information is purged from the cache. The default value is **30** seconds.

actimeo=*n* Sets **acregmin**, **acregmax**, **acdirmin**, and **acdirmax** to *n*.

EXAMPLES

The following example CacheFS-mounts the file system **server1:/user2**, which is already NFS-mounted on **/usr/abc** as **/xyz**.

```
example# mount -F cachefs -o backfstype=nfs,backpath=/usr/abc,
        cachedir=/cache1 server1:/user2 /xyz
```

The lines similar to the following appear in the **/etc/mnttab** file after the **mount** command is executed:

server1:/user2	**/usr/abc**	**nfs**	
/usr/abc	**/cache1/xyz**	**cachefs**	**backfstype=nfs**

SEE ALSO

cfsadmin(1M), **fsck_cachefs**(1M), **mount**(1M)
System Administration Guide, Volume I