NAME

cfsadmin – administer disk space used for caching file systems with the Cache File-System (CacheFS)

SYNOPSIS

cfsadmin –**c** [–**o** *cacheFS-parameters*] *cache_directory*
cfsadmin –**d** [*cache_ID* | **all**] *cache_directory*
cfsadmin –**l** *cache_directory*
cfsadmin –**s** [*mntpt1* . . . | **all**] *cache_directory*
cfsadmin –**u** [–**o** *cacheFS-parameters*]

AVAILABILITY

SUNWcsu

DESCRIPTION

The **cfsadmin** command provides the following functions:

- cache creation
- deletion of cached file systems
- listing of cache contents and statistics
- resource parameter adjustment when the file system is unmounted.

For each form of the command, with the except of the –**s** form, you must specify a cache directory, that is, the directory under which the cache is actually stored. A path name in the front file system identifies the cache directory. For the **-s** form of the command, you must specify a mount point.

You can specify a cache ID when you mount a file system with CacheFS, or you can let the system generate one for you. The –**l** option includes the cache ID in its listing of information. You must know the cache ID to delete a cached file system.

OPTIONS

–c Create a cache under the directory specified by *cache_directory*. This directory must not exist prior to cache creation.

–d Remove the file system whose cache ID you specify and release its resources, or remove all file systems in the cache by specifying **all**. After deleting a file system from the cache, you must run the **fsck_cachefs**(1M) command to correct the resource counts for the cache.

–l List file systems stored in the specified cache, as well as statistics about them. Each cached file system is listed by cache ID. The statistics document resource utilization and cache resource parameters.

–s Request a consistency check on the specified file system (or all cachefs mounted file systems). The –**s** option will only work if the cache file system was mounted with **demandconst** enabled (see **mount_cachefs**(1M)). Each file in the specified cache file system is checked for consistency with its corresponding file in the back file system. Note that the consistency check is performed file by file as files are accessed. If no files are accessed, no checks are performed. Use of this option will not result in a sudden "storm" of consistency checks.

–u Update resource parameters of the specified cache directory. Parameter values can only be increased. To decrease the values, you must remove the cache and recreate it. All file systems in the cache directory must be unmounted when you use this option. Changes will take effect the next time you mount any file system in the specified cache directory. The −**u** option with no −**o** option sets all parameters to their default values.

CacheFS Resource Parameters

You can specify the following cacheFS resource parameters as arguments to the −**o** option. Separate multiple parameters with commas.

maxblocks=*n* Maximum amount of storage space that CacheFS can use, expressed as a percentage of the total number of blocks in the front file system. If CacheFS does not have exclusive use of the front file system, there is no guarantee that all the space the **maxblocks** parameter allows will be available. The default is **90**.

minblocks=*n* Minimum amount of storage space, expressed as a percentage of the total number of blocks in the front file system, that CacheFS is always allowed to use without limitation by its internal control mechanisms. If CacheFS does not have exclusive use of the front file system, there is no guarantee that all the space the **minblocks** parameter attempts to reserve will be available. The default is **0**.

threshblocks=*n* A percentage of the total blocks in the front file system beyond which CacheFS cannot claim resources once its block usage has reached the level specified by **minblocks**. The default is **85**.

maxfiles=*n* Maximum number of files that CacheFS can use, expressed as a percentage of the total number of inodes in the front file system. If CacheFS does not have exclusive use of the front file system, there is no guarantee that all the inodes the **maxfiles** parameter allows will be available. The default is **90**.

minfiles=*n* Minimum number of files, expressed as a percentage of the total number of inodes in the front file system, that CacheFS is always allowed to use without limitation by its internal control mechanisms. If CacheFS does not have exclusive use of the front file system, there is no guarantee that all the inodes the **minfiles** parameter attempts to reserve will be available. The default is **0**.

threshfiles=*n* A percentage of the total inodes in the front file system beyond which CacheFS cannot claim inodes once its usage has reached the level specified by **minfiles**. The default is **85**.

maxfilesize=*n* Largest file size, expressed in megabytes, that CacheFS is allowed to cache. The default is **3**.

Note: You cannot decrease the block or inode allotment for a cache. To decrease the size of a cache, you must remove it and create it again with different parameters.

EXAMPLES

The following example creates a cache directory named **/cache**:

example# cfsadmin -c /cache

The following example creates a cache named **/cache1** that can claim a maximum of 60 percent of the blocks in the front file system, can use 40 percent of the front file system blocks without interference by CacheFS internal control mechanisms, and has a threshold value of 50 percent. The threshold value indicates that after CacheFS reaches its guaranteed minimum, it cannot claim more space if 50 percent of the blocks in the front file system are already used.

example# cfsadmin −c −o maxblocks=60,minblocks=40,
threshblocks=50 /cache1

The following example changes the **maxfilesize** parameter for the cache directory **/cache2** to 2 megabytes:

example# cfsadmin −u −o maxfilesize=2 /cache2

The following example lists the contents of a cache directory named **/cache3** and provides statistics about resource utilization:

example# cfsadmin −l /cache3

The following example removes the cached file system with cache ID **23** from the cache directory **/cache3** and frees its resources (the cache ID is part of the information returned by **cfsadmin −l**):

example# cfsadmin −d 23 /cache3

The following example removes all cached file systems from the cache directory **/cache3**:

example# cfsadmin −d all /cache3

The following example checks for consistency all filesystems mounted with **demandconst** enabled. No errors will be reported if no **demandconst** filesystems were found.

example# cfsadmin −s all

SEE ALSO

fsck_cachefs(1M), **mount_cachefs**(1M)