

new/usr/src/lib/libc/inc/mse.h

1

```
*****
2864 Wed Jun  3 21:39:38 2015
new/usr/src/lib/libc/inc/mse.h
5956 orientate is not a word
Reviewed by: Garrett D'Amore <garrett@damore.org>
Reviewed by: Marcel Telka <marcel@telka.sk>
*****
unchanged_portion_omitted_

47 /*
48 * DESCRIPTION:
49 * This function gets the pointer to the mbstate_t structure associated
50 * with the specified iop.
51 *
52 * RETURNS:
53 * If the associated mbstate_t found, the pointer to the mbstate_t is
54 * returned. Otherwise, (mbstate_t *)NULL is returned.
55 */
56 #ifdef _LP64
57 #define _getmbstate(iop)      (&(iop)->_state)
58 #else
59 extern mbstate_t      *_getmbstate(FILE *);
60 #endif

62 /*
63 * DESCRIPTION:
64 * This function/macro gets the orientation bound to the specified iop.
65 *
66 * RETURNS:
67 * _WC_MODE      if iop has been bound to Wide orientation
68 * _BYTE_MODE    if iop has been bound to Byte orientation
69 * _NO_MODE      if iop has been bound to neither Wide nor Byte
70 */
71 extern _IOP_orientation_t      _getorientation(FILE *);

73 /*
74 * DESCRIPTION:
75 * This function/macro sets the orientation to the specified iop.
76 *
77 * INPUT:
78 * flag may take one of the following:
79 *   _WC_MODE      Wide orientation
80 *   _BYTE_MODE    Byte orientation
81 *   _NO_MODE      Unoriented
82 */
83 extern void      _setorientation(FILE *, _IOP_orientation_t);

85 /*
86 * From page 32 of XSH5
87 * Once a wide-character I/O function has been applied
88 * to a stream without orientation, the stream becomes
89 * wide-oriented. Similarly, once a byte I/O function
90 * wide-orientated. Similarly, once a byte I/O function
91 * has been applied to a stream without orientation,
92 * the stream becomes byte-oriented. Only a call to
93 * the freopen() function or the fwide() function can
94 * otherwise alter the orientation of a stream.
95 */

96 #define _SET_ORIENTATION_BYTE(iop) \
97 { \
98     if (GET_NO_MODE(iop)) \
99         _setorientation(iop, _BYTE_MODE); \
100 }
unchanged_portion_omitted_
```

new/usr/src/man/man1m/mount_nfs.1m

1

27255 Wed Jun 3 21:39:38 2015

new/usr/src/man/man1m/mount_nfs.1m

5956 orientate is not a word

Reviewed by: Garrett D'Amore <garrett@damore.org>

Reviewed by: Marcel Telka <marcel@telka.sk>

```
1  \" te
2  \" Copyright (c) 2009 Sun Microsystems, Inc. All Rights Reserved
3  \" Copyright 1989 AT&T
4  \" The contents of this file are subject to the terms of the Common Development
5  \" See the License for the specific language governing permissions and limitati
6  \" fields enclosed by brackets \"[]\" replaced with your own identifying informat
7  .TH MOUNT_NFS 1M \"Jun 3, 2015\"
8  .SH MOUNT_NFS 1M \"Jul 26, 2009\"
9  .SH NAME
10 mount_nfs \- mount remote NFS resources
11 .SH SYNOPSIS
12 .LP
13 \fBmount_nfs [\fB-F\fR nfs] [\fIgeneric_options\fR] [\fB-o\fR \fIspecific_options
14 .fi

16 .LP
17 .nf
18 \fBmount_nfs [\fB-F\fR nfs] [\fIgeneric_options\fR] [\fB-o\fR \fIspecific_options
19 .fi

21 .LP
22 .nf
23 \fBmount_nfs [\fB-F\fR nfs] [\fIgeneric_options\fR] [\fB-o\fR \fIspecific_options
24 [\fB-O\fR] \fIresource\fR \fImount_point\fR
25 .fi

27 .SH DESCRIPTION
28 .sp
29 .LP
30 The \fBmount_nfs\fR utility attaches a named \fIresource\fR to the file system
31 hierarchy at the pathname location \fImount_point\fR, which must already exist.
32 If \fImount_point\fR has any contents prior to the \fBmount_nfs\fR operation, the
33 contents remain hidden until the \fIresource\fR is once again unmounted.
34 .sp
35 .LP
36 \fBmount_nfs\fR starts the \fBblockd\fR(1M) and \fBstatd\fR(1M) daemons if they
37 are not already running.
38 .sp
39 .LP
40 If the resource is listed in the \fB/etc/vfstab\fR file, the command line can
41 specify either \fIresource\fR or \fImount_point\fR, and \fBmount_nfs\fR consults
42 \fB/etc/vfstab\fR for more information. If the \fB-F\fR option is omitted,
43 \fBmount_nfs\fR takes the file system type from \fB/etc/vfstab\fR.
44 .sp
45 .LP
46 If the resource is not listed in the \fB/etc/vfstab\fR file, then the command
47 line must specify both the \fIresource\fR and the \fImount_point\fR.
48 .sp
49 .LP
50 \fIhost\fR can be an IPv4 or IPv6 address string. As IPv6 addresses already
51 contain colons, enclose \fIhost\fR in a pair of square brackets when specifying
52 an IPv6 address string. Otherwise the first occurrence of a colon can be
53 interpreted as the separator between the host name and path, for example,
54 \fB[1080::8:800:200C:417A]:tmp/file\fR. See \fBbinet\fR(7P) and \fBbinet6\fR(7P).
55 .sp
56 .ne 2
57 .na
58 \fB\fIhost\fR:\fIpathname\fR\fR
```

new/usr/src/man/man1m/mount_nfs.1m

2

```
59 .ad
60 .sp .6
61 .RS 4n
62 Where \fIhost\fR is the name of the \fBNFS\fR server host, and \fIpathname\fR
63 is the path name of the directory on the server being mounted. The path name is
64 interpreted according to the server's path name parsing rules and is not
65 necessarily slash-separated, though on most servers, this is the case.
66 .RE

68 .sp
69 .ne 2
70 .na
71 \fB\fIinfs\fR://\fIhost\fR[:\fIport\fR]/\fIpathname\fR\fR
72 .ad
73 .sp .6
74 .RS 4n
75 This is an \fBNFS\fR \fBURL\fR and follows the standard convention for
76 \fBNFS\fR \fBURLs\fR as described in \fINFOS URL Scheme\fR, RFC 2224. See the
77 discussion of \fBURLs\fR and the public option under \fBNFS FILE SYSTEMS\fR
78 for a more detailed discussion.
79 .RE

81 .sp
82 .ne 2
83 .na
84 \fB\fIhost\fR:\fIpathname\fR
85 \fB\fIinfs\fR://\fIhost\fR[:\fIport\fR]/\fIpathname\fR\fR
86 .ad
87 .br
88 .na
89 \fB\fR
90 .ad
91 .sp .6
92 .RS 4n
93 \fIhost\fR:\fIpathname\fR is a comma-separated list of
94 \fIhost\fR:\fIpathname\fR.
95 .sp
96 See the discussion of replicated file systems and failover under \fBNFS FILE
97 SYSTEMS\fR for a more detailed discussion.
98 .RE

100 .sp
101 .ne 2
102 .na
103 \fB\fIhostlist\fR \fIpathname\fR\fR
104 .ad
105 .sp .6
106 .RS 4n
107 \fIhostlist\fR is a comma-separated list of hosts.
108 .sp
109 See the discussion of replicated file systems and failover under \fBNFS FILE
110 SYSTEMS\fR for a more detailed discussion.
111 .RE

113 .sp
114 .LP
115 The \fBmount_nfs\fR command maintains a table of mounted file systems in
116 \fB/etc/mnttab\fR, described in \fBmnttab\fR(4).
117 .sp
118 .LP
119 \fBmount_nfs\fR supports both NFSv3 and NFSv4 mounts. The default NFS version
120 is NFSv4.
121 .SH OPTIONS
122 .sp
123 .LP
124 See \fBmount_nfs(1M)\fR for the list of supported \fIgeneric_options\fR. See
```

```

125 \fBshare_nfs\fR(1M) for a description of server options.
126 .sp
127 .ne 2
128 .na
129 \fB\fB-o\fR \fBIspecific_options\fR\fR
130 .ad
131 .sp .6
132 .RS 4n
133 Set file system specific options according to a comma-separated list with no
134 intervening spaces.
135 .sp
136 .ne 2
137 .na
138 \fB\fBacdirmax=\fR\fIn\fR\fR
139 .ad
140 .sp .6
141 .RS 4n
142 Hold cached attributes for no more than \fIn\fR seconds after directory update.
143 The default value is \fB60\fR.
144 .RE

146 .sp
147 .ne 2
148 .na
149 \fB\fBacdirmin=\fR\fIn\fR\fR
150 .ad
151 .sp .6
152 .RS 4n
153 Hold cached attributes for at least \fIn\fR seconds after directory update. The
154 default value is \fB30\fR.
155 .RE

157 .sp
158 .ne 2
159 .na
160 \fB\fBacregmax=\fR\fIn\fR\fR
161 .ad
162 .sp .6
163 .RS 4n
164 Hold cached attributes for no more than \fIn\fR seconds after file
165 modification. The default value is \fB60\fR.
166 .RE

168 .sp
169 .ne 2
170 .na
171 \fB\fBacregmin=\fR\fIn\fR\fR
172 .ad
173 .sp .6
174 .RS 4n
175 Hold cached attributes for at least \fIn\fR seconds after file modification.
176 The default value is \fB3\fR.
177 .RE

179 .sp
180 .ne 2
181 .na
182 \fB\fBactimeo=\fR\fIn\fR\fR
183 .ad
184 .sp .6
185 .RS 4n
186 Set \fBfmin\fR and \fBfimax\fR times for regular files and directories to \fIn\fR
187 seconds. See "File Attributes," below, for a description of the effect of
188 setting this option to \fB0\fR.
189 .sp
190 See "Specifying Values for Attribute Cache Duration Options," below, for a

```

```

191 description of how \fBacdirmax\fR, \fBacdirmin\fR, \fBacregmax\fR,
192 \fBacregmin\fR, and \fBactimeo\fR are parsed on a \fBmount\fR command line.
193 .RE

195 .sp
196 .ne 2
197 .na
198 \fB\fBbg\fR | \fBfg\fR\fR
199 .ad
200 .sp .6
201 .RS 4n
202 If the first attempt fails, retry in the background, or, in the foreground. The
203 default is \fBfg\fR.
204 .RE

206 .sp
207 .ne 2
208 .na
209 \fB\fBforcedirectio\fR | \fBnoforcedirectio\fR\fR
210 .ad
211 .sp .6
212 .RS 4n
213 If \fBforcedirectio\fR is specified, then for the duration of the mount, forced
214 direct \fBIO\fR is used. If the filesystem is mounted using
215 \fBforcedirectio\fR, data is transferred directly between client and server,
216 with no buffering on the client. If the filesystem is mounted using
217 \fBnoforcedirectio\fR, data is buffered on the client. \fBforcedirectio\fR is a
218 performance option that is of benefit only in large sequential data transfers.
219 The default behavior is \fBnoforcedirectio\fR.
220 .RE

222 .sp
223 .ne 2
224 .na
225 \fB\fBgripid\fR\fR
226 .ad
227 .sp .6
228 .RS 4n
229 By default, the \fBGID\fR associated with a newly created file obeys the System
230 V semantics; that is, the \fBGID\fR is set to the effective \fBGID\fR of the
231 calling process. This behavior can be overridden on a per-directory basis by
232 setting the set-GID bit of the parent directory; in this case, the \fBGID\fR of
233 a newly created file is set to the \fBGID\fR of the parent directory (see
234 \fBbopen\fR(2) and \fBbmkdir\fR(2)). Files created on file systems that are
235 mounted with the \fBgripid\fR option obeys \fBBSD\fR semantics independent of
236 whether the set-GID bit of the parent directory is set; that is, the \fBGID\fR
237 is unconditionally inherited from that of the parent directory.
238 .RE

240 .sp
241 .ne 2
242 .na
243 \fB\fBhard\fR | \fBsoft\fR\fR
244 .ad
245 .sp .6
246 .RS 4n
247 Continue to retry requests until the server responds (\fBhard\fR) or give up
248 and return an error (\fBsoft\fR). The default value is \fBhard\fR. Note that
249 NFSv4 clients do not support soft mounts.
250 .RE

252 .sp
253 .ne 2
254 .na
255 \fB\fBintr\fR | \fBnointr\fR\fR
256 .ad

```

```

257 .sp .6
258 .RS 4n
259 Allow (do not allow) keyboard interrupts to kill a process that is hung while
260 waiting for a response on a hard-mounted file system. The default is
261 \fB\bintr\fR, which makes it possible for clients to interrupt applications that
262 can be waiting for a remote mount.
263 .RE

265 .sp
266 .ne 2
267 .na
268 \fB\bnoac\fR\fR
269 .ad
270 .sp .6
271 .RS 4n
272 Suppress data and attribute caching. The data caching that is suppressed is the
273 write-behind. The local page cache is still maintained, but data copied into it
274 is immediately written to the server.
275 .RE

277 .sp
278 .ne 2
279 .na
280 \fB\bnocto\fR\fR
281 .ad
282 .sp .6
283 .RS 4n
284 Do not perform the normal close-to-open consistency. When a file is closed, all
285 modified data associated with the file is flushed to the server and not held on
286 the client. When a file is opened the client sends a request to the server to
287 validate the client's local caches. This behavior ensures a file's consistency
288 across multiple NFS clients. When \fB\bnocto\fR is in effect, the client does
289 not perform the flush on close and the request for validation, allowing the
290 possibility of differences among copies of the same file as stored on multiple
291 clients.
292 .sp
293 This option can be used where it can be guaranteed that accesses to a specified
294 file system are made from only one client and only that client. Under such a
295 condition, the effect of \fB\bnocto\fR can be a slight performance gain.
296 .RE

298 .sp
299 .ne 2
300 .na
301 \fB\bport=\fR\fIn\fR\fR
302 .ad
303 .sp .6
304 .RS 4n
305 The server \fB\bport\fR port number. The default is \fB\BNFS_PORT\fR. If the
306 \fB\bport\fR option is specified, and if the resource includes one or more
307 \fB\BNFS\fR \fB\BURL\fRs, and if any of the \fB\BURL\fRs include a \fB\bport\fR number,
308 then the \fB\bport\fR number in the option and in the \fB\BURL\fR must be the same.
309 .RE

311 .sp
312 .ne 2
313 .na
314 \fB\bposix\fR\fR
315 .ad
316 .sp .6
317 .RS 4n
318 Request \fB\BPOSIX.1\fR semantics for the file system. Requires a mount Version 2
319 \fB\bmountd\fR(1M) on the server. See \fB\Bstandards\fR(5) for information
320 regarding POSIX.
321 .RE

```

```

323 .sp
324 .ne 2
325 .na
326 \fB\bproto=\fR\fInetid\fR | \fB\brdma\fR\fR
327 .ad
328 .sp .6
329 .RS 4n
330 By default, the transport protocol that the NFS mount uses is the first
331 available RDMA transport supported both by the client and the server. If no
332 RDMA transport is found, then it attempts to use a TCP transport or, failing
333 that, a UDP transport, as ordered in the \fB/etc/netconfig\fR file. If it does
334 not find a connection oriented transport, it uses the first available
335 connectionless transport.
336 .sp
337 Use this option to override the default behavior.
338 .sp
339 \fB\bproto\fR is set to the value of \fB\finetid\fR or \fB\brdma\fR. \fB\finetid\fR is
340 the value of the \fB\Bnetwork_id\fR field entry in the \fB/etc/netconfig\fR file.
341 .sp
342 The UDP protocol is not supported for NFS Version 4. If you specify a UDP
343 protocol with the \fB\bproto\fR option, NFS version 4 is not used.
344 .RE

346 .sp
347 .ne 2
348 .na
349 \fB\bpublic\fR\fR
350 .ad
351 .sp .6
352 .RS 4n
353 The \fB\bpublic\fR option forces the use of the public file handle when
354 connecting to the \fB\BNFS\fR server. The resource specified might not have an
355 \fB\BNFS\fR \fB\BURL\fR. See the discussion of \fB\BURL\fRs and the public option
356 under \fB\BNFS FILE SYSTEMS\fR for a more detailed discussion.
357 .RE

359 .sp
360 .ne 2
361 .na
362 \fB\bquota\fR | \fB\bnoquota\fR\fR
363 .ad
364 .sp .6
365 .RS 4n
366 Enable or prevent \fB\bquota\fR(1M) to check whether the user is over quota on
367 this file system; if the file system has quotas enabled on the server, quotas
368 are still checked for operations on this file system.
369 .RE

371 .sp
372 .ne 2
373 .na
374 \fB\bremount\fR\fR
375 .ad
376 .sp .6
377 .RS 4n
378 Remounts a read-only file system as read-write (using the \fB\brw\fR option).
379 This option cannot be used with other \fB\b-o\fR options, and this option works
380 only on currently mounted read-only file systems.
381 .RE

383 .sp
384 .ne 2
385 .na
386 \fB\bbretrans=\fR\fIn\fR\fR
387 .ad
388 .sp .6

```

```

389 .RS 4n
390 Set the number of \fBNFS\fR retransmissions to \fIn\fR. The default value is
391 \fB5\fR. For connection-oriented transports, this option has no effect because
392 it is assumed that the transport performs retransmissions on behalf of NFS.
393 .RE

395 .sp
396 .ne 2
397 .na
398 \fB\fBretry=\fR\fIn\fR\fR
399 .ad
400 .sp .6
401 .RS 4n
402 The number of times to retry the \fBmount\fR operation. The default for the
403 \fBmount\fR command is \fB10000\fR.
404 .sp
405 The default for the automounter is \fB0\fR, in other words, do not retry. You
406 might find it useful to increase this value on heavily loaded servers, where
407 automounter traffic is dropped, causing unnecessary server not responding
408 errors.
409 .RE

411 .sp
412 .ne 2
413 .na
414 \fB\fBrsz=\fR\fIn\fR\fR
415 .ad
416 .sp .6
417 .RS 4n
418 Set the read buffer size to a maximum of \fIn\fR bytes. The default value is
419 \fB1048576\fR when using connection-oriented transports with Version 3 or
420 \fB1048576\fR when using connection-orientated transports with Version 3 or
421 Version 4 of the \fBNFS\fR protocol, and \fB32768\fR when using connection-less
422 transports. The default can be negotiated down if the server prefers a smaller
423 transfer size. "\fBRead\fR" operations may not necessarily use the maximum
424 buffer size. When using Version 2, the default value is \fB32768\fR for all
425 transports.
426 .RE

427 .sp
428 .ne 2
429 .na
430 \fB\fBsec=\fR\fImode\fR\fR
431 .ad
432 .sp .6
433 .RS 4n
434 Set the security \fImode\fR for \fBNFS\fR transactions. If \fBsec=\fR is not
435 specified, then the default action is to use \fBAUTH_SYS\fR over \fBNFS\fR
436 Version 2 mounts, use a user-configured default \fBauth\fR over NFS version 3
437 mounts, or to negotiate a mode over Version 4 mounts.
438 .sp
439 The preferred mode for NFS Version 3 mounts is the default mode specified in
440 \fB/etc/nfssec.conf\fR (see \fBnfssec.conf\fR(4)) on the client. If there is no
441 default configured in this file or if the server does not export using the
442 client's default mode, then the client picks the first mode that it supports in
443 the array of modes returned by the server. These alternatives are limited to
444 the security flavors listed in \fB/etc/nfssec.conf\fR.
445 .sp
446 NFS Version 4 mounts negotiate a security mode when the server returns an array
447 of security modes. The client attempts the mount with each security mode, in
448 order, until one is successful.
449 .sp
450 Only one mode can be specified with the \fBsec=\fR option. See \fBnfssec\fR(5)
451 for the available \fImode\fR options.
452 .RE

```

```

454 .sp
455 .ne 2
456 .na
457 \fB\fBsecure\fR\fR
458 .ad
459 .sp .6
460 .RS 4n
461 This option has been deprecated in favor of the \fBsec=\fR\fIdh\fR option.
462 .RE

464 .sp
465 .ne 2
466 .na
467 \fB\fBtimeo=\fR\fIn\fR\fR
468 .ad
469 .sp .6
470 .RS 4n
471 Set the \fBNFS\fR timeout to \fIn\fR tenths of a second. The default value is
472 \fB11\fR tenths of a second for connectionless transports, and \fB600\fR tenths
473 of a second for connection-oriented transports. This value is ignored for
474 connectionless transports. Such transports might implement their own timeouts,
475 which are outside the control of NFS.
476 .RE

478 .sp
479 .ne 2
480 .na
481 \fB\fBvers=\fR\fINFS version number\fR\fR
482 .ad
483 .sp .6
484 .RS 4n
485 By default, the version of \fBNFS\fR protocol used between the client and the
486 server is the highest one available on both systems. The default maximum for
487 the client is Version 4. This can be changed by setting the
488 \fBNFS_CLIENT_VERSION_MAX\fR parameter in \fB/etc/default/nfs\fR to a valid version
489 (2, 3, or 4). If the \fBNFS\fR server does not support the client's default
490 maximum, the next lowest version attempted until a matching version is found.
491 .RE

493 .sp
494 .ne 2
495 .na
496 \fB\fBwsize=\fR\fIn\fR\fR
497 .ad
498 .sp .6
499 .RS 4n
500 Set the write buffer size to a maximum of \fIn\fR bytes. The default value is
501 \fB1048576\fR when using connection-oriented transports with Version 3 or
502 \fB1048576\fR when using connection-orientated transports with Version 3 or
503 Version 4 of the \fBNFS\fR protocol, and \fB32768\fR when using connection-less
504 transports. The default can be negotiated down if the server prefers a smaller
505 buffer size. "\fBWrite\fR" operations may not necessarily use the maximum
506 buffer size. When using Version 2, the default value is \fB32768\fR for all
507 transports.
508 .RE

509 .sp
510 .ne 2
511 .na
512 \fB\fBxattr\fR | \fBnoxattr\fR\fR
513 .ad
514 .sp .6
515 .RS 4n
516 Allow or disallow the creation and manipulation of extended attributes. The
517 default is \fBxattr\fR. See \fBfsattr\fR(5) for a description of extended
518 attributes.

```

```

519 .RE
521 .RE

523 .sp
524 .ne 2
525 .na
526 \fB\fB-O\fR\fR
527 .ad
528 .sp .6
529 .RS 4n
530 Overlay mount. Allow the file system to be mounted over an existing mount
531 point, making the underlying file system inaccessible. If a mount is attempted
532 on a pre-existing mount point without setting this flag, the mount fails,
533 producing the error "device busy."
534 .RE

536 .SH NFS FILE SYSTEMS
537 .sp
538 .ne 2
539 .na
540 \fBBackground versus Foreground\fR
541 .ad
542 .sp .6
543 .RS 4n
544 File systems mounted with the \fBbg\fR option indicate that \fBmount\fR is to
545 retry in the background if the server's mount daemon (\fBmountd\fR(1M)) does
546 not respond. \fBmount\fR retries the request up to the count specified in the
547 \fBretry\fR option. (Note that the default value for \fBretry\fR
548 differs between \fBmount\fR and \fBautomount\fR. See the description of
549 \fBretry\fR, above.) Once the file system is mounted, each \fBNFS\fR request
550 made in the kernel waits \fBtimeo\fR tenths of a second for a response.
551 If no response arrives, the time-out is multiplied by \fB2\fR and the request
552 is retransmitted. When the number of retransmissions has reached the number
553 specified in the \fBretrans\fR option, a file system mounted with the
554 \fBsoft\fR option returns an error on the request; one mounted with the
555 \fBhard\fR option prints a warning message and continues to retry the request.
556 .RE

558 .sp
559 .ne 2
560 .na
561 \fBHard versus Soft\fR
562 .ad
563 .sp .6
564 .RS 4n
565 File systems that are mounted read-write or that contain executable files
566 should always be mounted with the \fBhard\fR option. Applications using
567 \fBsoft\fR mounted file systems can incur unexpected \fBI/O\fR errors, file
568 corruption, and unexpected program core dumps. The soft option is not
569 recommended.
570 .RE

572 .sp
573 .ne 2
574 .na
575 \fBAuthenticated requests\fR
576 .ad
577 .sp .6
578 .RS 4n
579 The server can require authenticated \fBNFS\fR requests from the client.
580 \fBsec\fR option might be required. See \fBnfssec\fR(5).
581 .RE

583 .sp
584 .ne 2

```

```

585 .na
586 \fBURLs and the public option\fR
587 .ad
588 .sp .6
589 .RS 4n
590 If the \fBpublic\fR option is specified, or if the \fBresource\fR includes and
591 \fBNFS\fR \fBURL\fR, \fBmount\fR attempts to connect to the server using the
592 public file handle lookup protocol. See \fIWebNFS Client Specification\fR, RFC
593 2054. If the server supports the public file handle, the attempt is successful;
594 \fBmount\fR does not need to contact the server's \fBbrpcbind\fR(1M) and the
595 \fBmountd\fR(1M) daemons to get the port number of the \fBmount\fR server and
596 the initial file handle of \fBipathname\fR, respectively. If the \fBNFS\fR
597 client and server are separated by a firewall that allows all outbound
598 connections through specific ports, such as \fBNFS_PORT\fR, then this enables
599 \fBNFS\fR operations through the firewall. The public option and the \fBNFS\fR
600 \fBURL\fR can be specified independently or together. They interact as
601 specified in the following matrix:
602 .sp
603 .in +2
604 .nf
605
606 Resource Style
607
608 \fIhost\fR:\fBipathname\fR NFS URL
609
610 public option Force public file Force public file
611 handle and fail handle and fail
612 mount if not supported. mount if not supported.
613
614 Use Native paths. Use Canonical paths.
615
616 default Use MOUNT protocol. Try public file handle
617 with Canonical paths.
618 Fall back to MOUNT
619 protocol if not
620 supported.
621 .fi
622 .in -2

623 A Native path is a path name that is interpreted according to conventions used
624 on the native operating system of the \fBNFS\fR server. A Canonical path is a
625 path name that is interpreted according to the \fBURL\fR rules. See \fIUniform
626 Resource Locators (URL)\fR, RFC 1738. See for uses of Native and Canonical
627 paths.
628 .RE

630 .sp
631 .ne 2
632 .na
633 \fBReplicated file systems and failover\fR
634 .ad
635 .sp .6
636 .RS 4n
637 \fBresource\fR can list multiple read\(\mionly file systems to be used to
638 provide data. These file systems should contain equivalent directory structures
639 and identical files. It is also recommended that they be created by a utility
640 such as \fBrdist\fR(1). The file systems can be specified either with a
641 comma\(\mimseparated list of \fIhost:/pathname\fR entries and/or \fBNFS\fR
642 \fBURL\fR entries, or with a comma\(\mimseparated list of hosts, if all file
643 system names are the same. If multiple file systems are named and the first
644 server in the list is down, failover uses the next alternate server to access
645 files. If the read\(\mionly option is not chosen, replication is disabled. File
646 access, for NFS Versions 2 and 3, is blocked on the original if NFS locks are
647 active for that file.
648 .RE

650 .SS "File Attributes"

```

```

651 .sp
652 .LP
653 To improve \fBNFS\fR read performance, files and file attributes are cached.
654 File modification times get updated whenever a write occurs. However, file
655 access times can be temporarily out-of-date until the cache gets refreshed.
656 .sp
657 .LP
658 The attribute cache retains file attributes on the client. Attributes for a
659 file are assigned a time to be flushed. If the file is modified before the
660 flush time, then the flush time is extended by the time since the last
661 modification (under the assumption that files that changed recently are likely
662 to change soon). There is a minimum and maximum flush time extension for
663 regular files and for directories. Setting \fBactimeo=\fR\fIn\fR sets flush
664 time to \fIn\fR seconds for both regular files and directories.
665 .sp
666 .LP
667 Setting \fBactimeo=0\fR disables attribute caching on the client. This means
668 that every reference to attributes is satisfied directly from the server though
669 file data is still cached. While this guarantees that the client always has the
670 latest file attributes from the server, it has an adverse effect on performance
671 through additional latency, network load, and server load.
672 .sp
673 .LP
674 Setting the \fBnoac\fR option also disables attribute caching, but has the
675 further effect of disabling client write caching. While this guarantees that
676 data written by an application is written directly to a server, where it can be
677 viewed immediately by other clients, it has a significant adverse effect on
678 client write performance. Data written into memory-mapped file pages
679 (\fBmmap\fR(2)) are not written directly to this server.
680 .SS "Specifying Values for Attribute Cache Duration Options"
681 .sp
682 .LP
683 The attribute cache duration options are \fBacdirmax\fR, \fBacdirmin\fR,
684 \fBacregmax\fR, \fBacregmin\fR, and \fBactimeo\fR, as described under OPTIONS.
685 A value specified for \fBactimeo\fR sets the values of all attribute cache
686 duration options except for any of these options specified following
687 \fBactimeo\fR on a \fBmount\fR command line. For example, consider the
688 following command:
689 .sp
690 .in +2
691 .nf
692 example# mount -o acdirmax=10,actimeo=1000 server:/path /localpath
693 .fi
694 .in -2

696 .sp
697 .LP
698 Because \fBactimeo\fR is the last duration option in the command line, its
699 value (\fB1000\fR) becomes the setting for all of the duration options,
700 including \fBacdirmax\fR. Now consider:
701 .sp
702 .in +2
703 .nf
704 example# mount -o actimeo=1000,acdirmax=10 server:/path /localpath
705 .fi
706 .in -2

708 .sp
709 .LP
710 Because the \fBacdirmax\fR option follows \fBactimeo\fR on the command line, it
711 is assigned the value specified (\fB10\fR). The remaining duration options are
712 set to the value of \fBactimeo\fR (\fB1000\fR).
713 .SH EXAMPLES
714 .LP
715 \fBExample 1 \fRMounting an \fBNFS\fR File System
716 .sp

```

```

717 .LP
718 To mount an \fBNFS\fR file system:

720 .sp
721 .in +2
722 .nf
723 example# mount serv:/usr/src /usr/src
724 .fi
725 .in -2
726 .sp

728 .LP
729 \fBExample 2 \fRMounting An \fBNFS\fR File System Read-Only With No suid
730 Privileges
731 .sp
732 .LP
733 To mount an \fBNFS\fR file system read-only with no suid privileges:

735 .sp
736 .in +2
737 .nf
738 example# mount -r -o nosuid serv:/usr/src /usr/src
739 .fi
740 .in -2
741 .sp

743 .LP
744 \fBExample 3 \fRMounting An \fBNFS\fR File System Over Version 2, with the UDP
745 Transport
746 .sp
747 .LP
748 To mount an \fBNFS\fR file system over Version 2, with the UDP transport:

750 .sp
751 .in +2
752 .nf
753 example# mount -o vers=2,proto=udp serv:/usr/src /usr/src
754 .fi
755 .in -2
756 .sp

758 .LP
759 \fBExample 4 \fRMounting an \fBNFS\fR File System Using An \fBNFS\fR \fBURL\fR
760 .sp
761 .LP
762 To mount an \fBNFS\fR file system using an \fBNFS\fR \fBURL\fR (a canonical
763 path):

765 .sp
766 .in +2
767 .nf
768 example# mount nfs://serv/usr/man /usr/man
769 .fi
770 .in -2
771 .sp

773 .LP
774 \fBExample 5 \fRMounting An \fBNFS\fR File System Forcing Use Of The Public
775 File Handle
776 .sp
777 .LP
778 To mount an \fBNFS\fR file system and force the use of the public file handle
779 and an \fBNFS\fR \fBURL\fR (a canonical path) that has a non 7-bit ASCII escape
780 sequence:

782 .sp

```

```

783 .in +2
784 .nf
785 example# mount -o public nfs://serv/usr/%A0abc /mnt/test
786 .fi
787 .in -2
788 .sp

790 .LP
791 \fBExample 6 \fRMounting an \fBNFS\fR File System Using a Native Path
792 .sp
793 .LP
794 To mount an \fBNFS\fR file system using a native path (where the server uses
795 colons (":") as the component separator) and the public file handle:

797 .sp
798 .in +2
799 .nf
800 example# mount -o public serv:C:doc:new /usr/doc
801 .fi
802 .in -2
803 .sp

805 .LP
806 \fBExample 7 \fRMounting a Replicated Set of \fBNFS\fR File Systems with the
807 Same Pathnames
808 .sp
809 .LP
810 To mount a replicated set of \fBNFS\fR file systems with the same pathnames:

812 .sp
813 .in +2
814 .nf
815 example# mount serv\(mia,serv\(mib,serv\(mic:/usr/man /usr/man
816 .fi
817 .in -2
818 .sp

820 .LP
821 \fBExample 8 \fRMounting a Replicated Set of \fBNFS\fR File Systems with
822 Different Pathnames
823 .sp
824 .LP
825 To mount a replicated set of \fBNFS\fR file systems with different pathnames:

827 .sp
828 .in +2
829 .nf
830 example# mount serv\(mix:/usr/man,serv\(miy:/var/man,nfs://serv-z/man /usr/man
831 .fi
832 .in -2
833 .sp

835 .SH FILES
836 .sp
837 .ne 2
838 .na
839 \fB\fB/etc/mnttab\fR\fR
840 .ad
841 .sp .6
842 .RS 4n
843 table of mounted file systems
844 .RE

846 .sp
847 .ne 2
848 .na

```

```

849 \fB\fB/etc/dfs/fstypes\fR\fR
850 .ad
851 .sp .6
852 .RS 4n
853 default distributed file system type
854 .RE

856 .sp
857 .ne 2
858 .na
859 \fB\fB/etc/vfstab\fR\fR
860 .ad
861 .sp .6
862 .RS 4n
863 table of automatically mounted resources
864 .RE

866 .SH SEE ALSO
867 .sp
868 .LP
869 \fBBrdist\fR(1), \fBBlockd\fR(1M), \fBMountall\fR(1M), \fBMountd\fR(1M),
870 \fBbnfsd\fR(1M), \fBbquota\fR(1M), \fBbstatd\fR(1M), \fBbmkdir\fR(2),
871 \fBbmmap\fR(2), \fBbmount\fR(2), \fBbopen\fR(2), \fBbumount\fR(2), \fBbmnttab\fR(4),
872 \fBbnfs\fR(4), \fBbnfssec.conf\fR(4), \fBbattributes\fR(5), \fBbfsattr\fR(5),
873 \fBbnfssec\fR(5), \fBbstandards\fR(5), \fBbinet\fR(7P), \fBbinet6\fR(7P),
874 \fBblofs\fR(7FS)
875 .sp
876 .LP
877 Callaghan, Brent, \fIWebNFS Client Specification\fR, RFC 2054, October 1996.
878 .sp
879 .LP
880 Callaghan, Brent, \fIINFS URL Scheme\fR, RFC 2224, October 1997.
881 .sp
882 .LP
883 Berners-Lee, Masinter & McCahill, \fIUniform Resource Locators (URL)\fR, RFC
884 1738, December 1994.
885 .SH NOTES
886 .sp
887 .LP
888 An \fBNFS\fR server should not attempt to mount its own file systems. See
889 \fBblofs\fR(7FS).
890 .sp
891 .LP
892 If the directory on which a file system is to be mounted is a symbolic link,
893 the file system is mounted on \fBthe directory to which the symbolic link
894 refers,\fR rather than being mounted on top of the symbolic link itself.
895 .sp
896 .LP
897 SunOS 4.x used the \fBbiiod\fR maintenance procedure to perform parallel
898 read-ahead and write-behind on \fBNFS\fR clients. SunOS 5.x made \fBbiiod\fR
899 obsolete with multi-threaded processing, which transparently performs parallel
900 read-ahead and write-behind.
901 .sp
902 .LP
903 Since the root \fB(\fR\fB)\fR file system is mounted read-only by the kernel
904 during the boot process, only the \fBRemount\fR option (and options that can be
905 used in conjunction with \fBRemount\fR) affect the root (\fB/\fR) entry in the
906 \fB/etc/vfstab\fR file.
907 .sp
908 .LP
909 \fBmount_cache\fR cannot be used with replicated NFS mounts or any NFS
910 Version 4 mount.
911 .sp
912 .LP
913 The NFS client service is managed by the service management facility,
914 \fBsmf\fR(5), under the service identifier:

```


new/usr/src/man/man1m/mount_nfs.1m

15

```
915 .sp
916 .in +2
917 .nf
918 svc:/network/nfs/client:default
919 .fi
920 .in -2
921 .sp
```

```
923 .sp
924 .LP
925 Administrative actions on this service, such as enabling, disabling, or
926 requesting restart, can be performed using \fBsvcadm\fR(1M). The service's
927 status can be queried using the \fBsvcs\fR(1) command.
```

```

*****
3839 Wed Jun  3 21:39:39 2015
new/usr/src/man/man3c/fwide.3c
5956 orientate is not a word
Reviewed by: Garrett D'Amore <garrett@damore.org>
Reviewed by: Marcel Telka <marcel@telka.sk>
*****
1  \" te
2  \" Copyright (c) 1992, X/Open Company Limited All Rights Reserved Portions Co
3  \" Sun Microsystems, Inc. gratefully acknowledges The Open Group for permission
4  \" http://www.opengroup.org/bookstore/.
5  \" The Institute of Electrical and Electronics Engineers and The Open Group, ha
6  \" This notice shall appear on any product containing this material.
7  \" The contents of this file are subject to the terms of the Common Development
8  \" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
9  \" When distributing Covered Code, include this CDDL HEADER in each file and in
10 .TH FWIDE 3C \"Jun 3, 2015\"
10 .TH FWIDE 3C \"Jul 24, 2002\"
11 .SH NAME
12 fwide \- set stream orientation
13 .SH SYNOPSIS
14 .LP
15 .nf
16 #include <stdio.h>
17 #include <wchar.h>

19 \fBint\fR \fBfwide\fR(\fBFILE *\fR \fIstream\fR, \fBint\fR \fImode\fR);
20 .fi

22 .SH DESCRIPTION
23 .sp
24 .LP
25 The \fBfwide()\fR function determines the orientation of the stream pointed to
26 by \fIstream\fR. If \fImode\fR is greater than 0, the function first attempts
27 to make the stream wide-oriented. If \fImode\fR is less than 0, the function
28 first attempts to make the stream byte-oriented. Otherwise, \fImode\fR is 0
29 to make the stream wide-oriented. If \fImode\fR is less than 0, the function
30 first attempts to make the stream byte-oriented. Otherwise, \fImode\fR is 0
31 and the function does not alter the orientation of the stream.
32 .sp
33 .LP
34 If the orientation of the stream has already been determined, \fBfwide()\fR
35 does not change it.
36 .sp
37 .LP
38 Because no return value is reserved to indicate an error, an application
39 wishing to check for error situations should set \fBerrno\fR to 0, then call
40 \fBfwide()\fR, then check \fBerrno\fR and if it is non-zero, assume an error
41 has occurred.
42 .SH RETURN VALUES
43 .sp
44 .LP
45 The \fBfwide()\fR function returns a value greater than 0 if, after the call,
46 the stream has wide-orientation, a value less than 0 if the stream has
47 byte-orientation, or 0 if the stream has no orientation.
48 .SH ERRORS
49 .sp
50 .LP
51 The \fBfwide()\fR function may fail if:
52 .sp
53 .ne 2
54 .na
55 \fBFBFBEBADF\fR \fR
56 .ad
57 .RS 9n
58 The \fIstream\fR argument is not a valid stream.

```

```

57 .RE

59 .SH USAGE
60 .sp
61 .LP
62 A call to \fBfwide()\fR with \fImode\fR set to 0 can be used to determine the
63 current orientation of a stream.
64 .SH ATTRIBUTES
65 .sp
66 .LP
67 See \fBattributes\fR(5) for descriptions of the following attributes:
68 .sp

70 .sp
71 .TS
72 box;
73 c | c
74 l | l .
75 ATTRIBUTE TYPE ATTRIBUTE VALUE
76 _
77 Interface Stability Standard
78 _
79 MT-Level MT-Safe
80 .TE

82 .SH SEE ALSO
83 .sp
84 .LP
85 \fBattributes\fR(5), \fBstandards\fR(5)

```

new/usr/src/uts/common/sys/t_kuser.h

1

```
*****
4405 Wed Jun  3 21:39:39 2015
new/usr/src/uts/common/sys/t_kuser.h
5956 orientate is not a word
Reviewed by: Garrett D'Amore <garrett@damore.org>
Reviewed by: Marcel Telka <marcel@telka.sk>
*****

1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License, Version 1.0 only
6  * (the "License").  You may not use this file except in compliance
7  * with the License.
8  *
9  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 * or http://www.opensolaris.org/os/licensing.
11 * See the License for the specific language governing permissions
12 * and limitations under the License.
13 *
14 * When distributing Covered Code, include this CDDL HEADER in each
15 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 * If applicable, add the following below this CDDL HEADER, with the
17 * fields enclosed by brackets "[]" replaced with your own identifying
18 * information: Portions Copyright [yyyy] [name of copyright owner]
19 *
20 * CDDL HEADER END
21 */
22 /*
23  * Copyright 2014 Garrett D'Amore <garrett@damore.org>
24  *
25  * Copyright 1998 Sun Microsystems, Inc.  All rights reserved.
26  * Use is subject to license terms.
27  */

29 /*      Copyright (c) 1983, 1984, 1985, 1986, 1987, 1988, 1989 AT&T      */
30 /*      All Rights Reserved      */

32 /*
33  * University Copyright- Copyright (c) 1982, 1986, 1988
34  * The Regents of the University of California
35  * All Rights Reserved
36  *
37  * University Acknowledgment- Portions of this document are derived from
38  * software developed by the University of California, Berkeley, and its
39  * contributors.
40  */

42 #ifndef _SYS_T_KUSER_H
43 #define _SYS_T_KUSER_H

45 #include <sys/types.h>
46 #include <sys/file.h>
47 #include <sys/cred.h>
48 #include <sys/stream.h>
49 #include <sys/tiuser.h>

51 #ifdef __cplusplus
52 extern "C" {
53 #endif

55 /*
56  * Note this structure will need to be expanded to handle data
57  * related to connection oriented transports.
58  * related to connection orientated transports.
59  */
```

new/usr/src/uts/common/sys/t_kuser.h

2

```
59 typedef struct tiuser {
60     struct file *fp;
61     struct t_info tp_info; /* Transport provider Info. */
62     int flags;
63 } TIUSER;
unchanged portion omitted
```