
7461 Wed Jan 15 13:15:35 2020

new/usr/src/man/man3pool/pool_associate.3pool

12198 typo paramter in some section 3pool man pages

```

1  \" te
2  .\" Copyright (c) 2005, Sun Microsystems, Inc. All Rights Reserved.
3  .\" The contents of this file are subject to the terms of the Common Development
4  .\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
5  .\" When distributing Covered Code, include this CDDL HEADER in each file and in
6  .TH POOL_ASSOCIATE 3POOL \"January 15, 2020\"
7  .TH POOL_ASSOCIATE 3POOL \"April 9, 2016\"
8  .SH NAME
9  pool_associate, pool_create, pool_destroy, pool_dissociate, pool_info,
10 pool_query_pool_resources \- resource pool manipulation functions
11 .SH SYNOPSIS
12 cc [ \fIfIflag\fR&.\|. \. ] \fIfIfile\fR&.\|. \. \fB-lpool\fR [ \fIfIlibrary\fR&.\|. \.
13 #include <pool.h>
14
15 \fBint\fR \fBpool_associate\fR(\fBpool_conf_t * \fR\fIconf\fR, \fBpool_t * \fR\fI
16 \fBpool_resource_t * \fR\fIresource\fR);
17 .fi
18
19 .LP
20 .nf
21 \fBpool_t * \fR\fBpool_create\fR(\fBpool_conf_t * \fR\fIconf\fR, \fBconst char * \fR
22 .fi
23
24 .LP
25 .nf
26 \fBint\fR \fBpool_destroy\fR(\fBpool_conf_t * \fR\fIconf\fR, \fBpool_t * \fR\fIpool
27 .fi
28
29 .LP
30 .nf
31 \fBint\fR \fBpool_dissociate\fR(\fBpool_conf_t * \fR\fIconf\fR, \fBpool_t * \fR\fI
32 \fBpool_resource_t * \fR\fIresource\fR);
33 .fi
34
35 .LP
36 .nf
37 \fBconst char * \fR\fBpool_info\fR(\fBpool_conf_t * \fR\fIconf\fR, \fBpool_t * \fR
38 \fBint\fR \fBiflags\fR);
39 .fi
40
41 .LP
42 .nf
43 \fBpool_resource_t * * \fR\fBpool_query_pool_resources\fR(\fBpool_conf_t * \fR\fIco
44 \fBpool_t * \fR\fIpool\fR, \fBuint_t * \fR\fIelem\fR, \fBpool_value_t * * \fR
45 .fi
46
47 .SH DESCRIPTION
48 These functions provide mechanisms for constructing and modifying pools entries
49 within a target pools configuration. The \fIconf\fR argument for each function
50 refers to the target configuration to which the operation applies.
51 .sp
52 .LP
53 The \fBpool_associate()\fR function associates the specified resource with
54 \fIpool\fR. A resource can be associated with multiple pools at the same time.
55 Any resource of this type that was formerly associated with this pool is no
56 longer associated with the pool. The new association replaces the earlier one.
57 .sp
58 .LP

```

```

59 The \fBpool_create()\fR function creates a new pool with the supplied name with
60 its default properties initialized, and associated with the default resource of
61 each type.
62 .sp
63 .LP
64 The \fBpool_destroy()\fR destroys the given pool association.
65 Associated resources are not modified.
66 .sp
67 .LP
68 The \fBpool_dissociate()\fR function removes the association between the given
69 resource and pool. On successful completion, the pool is associated with the
70 default resource of the same type.
71 .sp
72 .LP
73 The \fBpool_info()\fR function returns a string describing the given pool. The
74 string is allocated with \fBmalloc\fR(3C). The caller is responsible for freeing
75 the returned string. If the \fBiflags\fR option is non-zero, the string returned
76 also describes the associated resources of the pool.
77 .sp
78 .LP
79 The \fBpool_query_pool_resources()\fR function returns a null-terminated array
80 of resources currently associated with the pool that match the given list of
81 properties. The return value must be freed by the caller. The \fBinelem\fR
82 argument is set to be the length of the array returned.
83 .SH RETURN VALUES
84 .LP
84 Upon successful completion, \fBpool_create()\fR returns a new initialized pool.
85 Otherwise it returns \fBINULL\fR and \fBpool_error\fR(3POOL) returns the
86 pool-specific error value.
87 .sp
88 .LP
89 Upon successful completion, \fBpool_associate()\fR, \fBpool_destroy()\fR, and
90 \fBpool_dissociate()\fR return 0. Otherwise, they return -1 and
91 \fBpool_error()\fR returns the pool-specific error value.
92 .sp
93 .LP
94 Upon successful completion, \fBpool_info()\fR returns a string describing the
95 given pool. Otherwise it returns \fBINULL\fR and \fBpool_error()\fR returns the
96 pool-specific error value.
97 .sp
98 .LP
99 Upon successful completion, \fBpool_query_pool_resources()\fR returns a
100 null-terminated array of resources. Otherwise it returns \fBINULL\fR and
101 \fBpool_error()\fR returns the pool-specific error value.
102 .SH ERRORS
103 .LP
103 The \fBpool_create()\fR function will fail if:
104 .sp
105 .ne 2
106 .na
107 \fBFBPOE_BADPARAM\fR
108 .ad
109 .RS 20n
110 The supplied configuration's status is not \fBPOF_VALID\fR or \fBiname\fR is
111 already in use.
112 .RE
113
114 .sp
115 .ne 2
116 .na
117 \fBFBPOE_SYSTEM\fR
118 .ad
119 .RS 20n
120 A system error has occurred. Check the system error code for more details.
121 .RE

```

```

123 .sp
124 .ne 2
125 .na
126 \fb\fbPOE_INVALID_CONF\fr\fr
127 .ad
128 .RS 20n
129 The pool element could not be created because the configuration would be
130 invalid.
131 .RE

133 .sp
134 .ne 2
135 .na
136 \fb\fbPOE_PUTPROP\fr\fr
137 .ad
138 .RS 20n
139 One of the supplied properties could not be set.
140 .RE

142 .sp
143 .LP
144 The \fbpool_destroy()\fr function will fail if:
145 .sp
146 .ne 2
147 .na
148 \fb\fbPOE_BADPARAM\fr\fr
149 .ad
150 .RS 16n
151 The supplied configuration's status is not \fbPOF_VALID\fr.
152 .RE

154 .sp
155 .LP
156 The \fbpool_associate()\fr function will fail if:
157 .sp
158 .ne 2
159 .na
160 \fb\fbPOE_BADPARAM\fr\fr
161 .ad
162 .RS 16n
163 The supplied configuration's status is not \fbPOF_VALID\fr or the parameters
164 are supplied from a different configuration.
165 .RE

167 .sp
168 .ne 2
169 .na
170 \fb\fbPOE_SYSTEM\fr\fr
171 .ad
172 .RS 16n
173 A system error has occurred. Check the system error code for more details.
174 .RE

176 .sp
177 .LP
178 The \fbpool_disassociate()\fr function will fail if:
179 .sp
180 .ne 2
181 .na
182 \fb\fbPOE_BADPARAM\fr\fr
183 .ad
184 .RS 20n
185 The supplied configuration's status is not \fbPOF_VALID\fr or the parameters
186 are supplied from a different configuration.
187 .RE

```

```

189 .sp
190 .ne 2
191 .na
192 \fb\fbPOE_INVALID_CONF\fr\fr
193 .ad
194 .RS 20n
195 No resources could be located for the supplied configuration or the supplied
196 configuration is not valid (for example, more than one default for a resource
197 type was found.)
198 .RE

200 .sp
201 .ne 2
202 .na
203 \fb\fbPOE_SYSTEM\fr\fr
204 .ad
205 .RS 20n
206 A system error has occurred. Check the system error code for more details.
207 .RE

209 .sp
210 .LP
211 The \fbpool_info()\fr function will fail if:
212 .sp
213 .ne 2
214 .na
215 \fb\fbPOE_BADPARAM\fr\fr
216 .ad
217 .RS 20n
218 The supplied configuration's status is not \fbPOF_VALID\fr or the \fiflags\fr
219 parameter is neither 0 or 1.
220 parameter is neither 0 or 1.
221 .RE

222 .sp
223 .ne 2
224 .na
225 \fb\fbPOE_INVALID_CONF\fr\fr
226 .ad
227 .RS 20n
228 The configuration is invalid.
229 .RE

231 .sp
232 .ne 2
233 .na
234 \fb\fbPOE_SYSTEM\fr\fr
235 .ad
236 .RS 20n
237 A system error has occurred. Check the system error code for more details.
238 .RE

240 .sp
241 .LP
242 The \fbpool_query_pool_resources()\fr function will fail if:
243 .sp
244 .ne 2
245 .na
246 \fb\fbPOE_BADPARAM\fr\fr
247 .ad
248 .RS 20n
249 The supplied configuration's status is not \fbPOF_VALID\fr.
250 .RE

252 .sp
253 .ne 2

```

```
254 .na
255 \fB\fBPOE_INVALID_CONF\fR\fR
256 .ad
257 .RS 20n
258 The configuration is invalid.
259 .RE

261 .sp
262 .ne 2
263 .na
264 \fB\fBPOE_SYSTEM\fR\fR
265 .ad
266 .RS 20n
267 A system error has occurred. Check the system error code for more details.
268 .RE

270 .SH USAGE
275 .LP
271 Pool names are unique across pools in a given configuration file. It is an
272 error to attempt to create a pool with a name that is currently used by another
273 pool within the same configuration.
274 .SH ATTRIBUTES
280 .LP
275 See \fBattributes\fR(5) for descriptions of the following attributes:
276 .sp

278 .sp
279 .TS
280 box;
281 c | c
282 l | l .
283 ATTRIBUTE TYPE ATTRIBUTE VALUE
284 -
285 CSI Enabled
286 -
287 Interface Stability Unstable
288 -
289 MT-Level Safe
290 .TE

292 .SH SEE ALSO
299 .LP
293 \fBblibpool\fR(3LIB), \fBpool_error\fR(3POOL), \fBattributes\fR(5)
```

```

*****
3448 Wed Jan 15 13:15:35 2020
new/usr/src/man/man3pool/pool_component_info.3pool
12198 typo paramter in some section 3pool man pages
*****
1 \" te
2 .\" Copyright (c) 2003, Sun Microsystems, Inc. All Rights Reserved.
3 .\" The contents of this file are subject to the terms of the Common Development
4 .\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
5 .\" When distributing Covered Code, include this CDDL HEADER in each file and in
6 .TH POOL_COMPONENT_INFO 3POOL \"January 15, 2020\"
7 .TH POOL_COMPONENT_INFO 3POOL \"April 9, 2016\"
8 .SH NAME
9 pool_component_info, pool_get_owing_resource - resource pool component
10 functions
11 .SH SYNOPSIS
12 .LP
13 cc [ \fIflag\fR&.\|. \. ] \fIfile\fR&.\|. \. \fB-lpool\fR [ \fIlibrary\fR&.\|. \. ]
14 #include <pool.h>
15 \fBconst char * \fR \fBpool_component_info\fR(\fBpool_conf_t * \fR \fBiconf\fR,
16 \fBpool_component_t * \fR \fBicomponent\fR, \fBbint\fR \fBiflags\fR);
17 .fi
18 .sp
19 .LP
20 .nf
21 \fBpool_resource_t * \fR \fBpool_get_owing_resource\fR(\fBpool_conf_t * \fR \fBiconf
22 \fBpool_component_t * \fR \fBicomponent\fR);
23 .fi
24 .sp
25 .SH DESCRIPTION
26 .LP
27 Certain resources, such as processor sets, are composed of resource components.
28 Informational and ownership attributes of resource components are made
29 available with the \fBpool_component_info()\fR and
30 \fBpool_get_owing_resource()\fR functions. The \fBiconf\fR argument for each
31 function refers to the target configuration to which the operation applies.
32 .sp
33 .LP
34 The \fBpool_component_info()\fR function returns a string describing
35 \fBicomponent\fR. The string is allocated with \fBmalloc\fR(3C). The caller is
36 responsible for freeing the returned string. The \fBiflags\fR argument is
37 ignored.
38 .sp
39 .LP
40 The \fBpool_get_owing_resource()\fR function returns the resource currently
41 containing \fBicomponent\fR. Every component is contained by a resource.
42 .SH RETURN VALUES
43 .LP
44 Upon successful completion, \fBpool_component_info()\fR returns a string.
45 Otherwise it returns \fBINULL\fR and \fBpool_error\fR(3POOL) returns the
46 pool-specific error value.
47 .sp
48 .LP
49 Upon successful completion, \fBpool_get_owing_resource()\fR returns the owing
50 resource. Otherwise it returns \fBINULL\fR and \fBpool_error()\fR returns the
51 pool-specific error value.
52 .SH ERRORS
53 .LP
54 The \fBpool_component_info()\fR function will fail if:
55 .sp
56 .ne 2
57 .na
58 \fBFBPOE_BADPARAM\fR
59 .ad

```

```

57 .RS 20n
58 The supplied configuration's status is not \fBPOF_VALID\fR or the \fBiflags\fR
59 parameter is neither 0 or 1.
60 paramter is neither 0 or 1.
61 .RE
62 .sp
63 .ne 2
64 .na
65 \fBFBPOE_INVALID_CONF\fR
66 .ad
67 .RS 20n
68 The configuration is invalid.
69 .RE
70 .sp
71 .sp
72 .ne 2
73 .na
74 \fBFBPOE_SYSTEM\fR
75 .ad
76 .RS 20n
77 A system error has occurred. Check the system error code for more details.
78 .RE
79 .sp
80 .LP
81 .LP
82 The \fBpool_get_owing_resource()\fR function will fail if:
83 .sp
84 .ne 2
85 .na
86 \fBFBPOE_BADPARAM\fR
87 .ad
88 .RS 16n
89 The supplied configuration's status is not \fBPOF_VALID\fR.
90 .RE
91 .sp
92 .SH ATTRIBUTES
93 .LP
94 See \fBattributes\fR(5) for descriptions of the following attributes:
95 .sp
96 .sp
97 .TS
98 box;
99 c | c
100 l | l .
101 ATTRIBUTE TYPE ATTRIBUTE VALUE
102 _
103 CSI Enabled
104 _
105 Interface Stability Unstable
106 _
107 MT-Level Safe
108 .TE
109 .sp
110 .SH SEE ALSO
111 .LP
112 \fBblibpool\fR(3LIB), \fBpool_error\fR(3POOL), \fBattributes\fR(5)

```

```

*****
8410 Wed Jan 15 13:15:35 2020
new/usr/src/man/man3pool/pool_resource_create.3pool
12198 typo paramter in some section 3pool man pages
*****
1 \" te
2 .\" Copyright (c) 2003, Sun Microsystems, Inc. All Rights Reserved.
3 .\" The contents of this file are subject to the terms of the Common Development
4 .\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
5 .\" When distributing Covered Code, include this CDDL HEADER in each file and in
6 .TH POOL_RESOURCE_CREATE 3POOL \"January 15, 2020\"
7 .TH POOL_RESOURCE_CREATE 3POOL \"April 9, 2016\"
8 .SH NAME
9 pool_resource_create, pool_resource_destroy, pool_resource_info,
10 pool_query_resource_components, pool_resource_transfer, pool_resource_xtransfer
11 \- resource pool resource manipulation functions
12 .SH SYNOPSIS
13 .LP
14 .nf
15 cc [ \fIflag\fR&.\|. \. ] \fIfile\fR&.\|. \. \fB-lpool\fR [ \fIlibrary\fR&.\|.
16 #include <pool.h>
17
18 \fBpool_resource_t * \fR \fBpool_resource_create\fR(\fBpool_conf_t * \fR \fIconf\fR,
19 \fBconst char * \fR \fItype\fR, \fBconst char * \fR \fIname\fR);
20 .fi
21
22 .LP
23 .nf
24 \fBint \fR \fBpool_resource_destroy\fR(\fBpool_conf_t * \fR \fIconf\fR,
25 \fBpool_resource_t * \fR \fIresource\fR);
26 .fi
27
28 .LP
29 .nf
30 \fBconst char * \fR \fBpool_resource_info\fR(\fBpool_conf_t * \fR \fIconf\fR
31 \fBpool_resource_t * \fR \fIresource\fR, \fBint \fR \fIflags\fR);
32 .fi
33
34 .LP
35 .nf
36 \fBpool_component_t ** \fR \fBpool_query_resource_components\fR(
37 \fBpool_conf_t * \fR \fIconf\fR, \fBpool_resource_t * \fR \fIresource\fR,
38 \fBuint_t * \fR \fIelem\fR, \fBpool_value_t ** \fR \fIprops\fR);
39 .fi
40
41 .LP
42 .nf
43 \fBint \fR \fBpool_resource_transfer\fR(\fBpool_conf_t * \fR \fIconf\fR,
44 \fBpool_resource_t * \fR \fIsource\fR, \fBpool_resource_t * \fR \fItarget\fR,
45 \fBuint64_t \fR \fIsize\fR);
46 .fi
47
48 .LP
49 .nf
50 \fBint \fR \fBpool_resource_xtransfer\fR(\fBpool_conf_t * \fR \fIconf\fR,
51 \fBpool_resource_t * \fR \fIsource\fR, \fBpool_resource_t * \fR \fItarget\fR,
52 \fBpool_component_t ** \fR \fIcomponents\fR);
53 .fi
54
55 .SH DESCRIPTION
56 .LP
57 The \fBpool_resource_create()\fR function creates and returns a new resource of
58 the given \fIname\fR and \fItype\fR in the provided configuration. If there is
59 already a resource of the given name, the operation will fail.
59 .sp
60 .LP

```

```

59 The \fBpool_resource_destroy()\fR function removes the specified \fIresource\fR
60 from its configuration file.
61 .sp
62 .LP
63 The \fBpool_resource_info()\fR function returns a string describing the given
64 \fIresource\fR. The string is allocated with \fBmalloc\fR(3C). The caller is
65 responsible for freeing the returned string. If the \fIflags\fR argument is
66 non-zero, the string returned also describes the components (if any) contained
67 in the resource.
68 .sp
69 .LP
70 The \fBpool_query_resource_components()\fR function returns a null-terminated
71 array of the components (if any) that comprise the given resource.
72 .sp
73 .LP
74 The \fBpool_resource_transfer()\fR function transfers \fIsize\fR basic units
75 from the \fIsource\fR resource to the \fItarget\fR. Both resources must be of
76 the same type for the operation to succeed. Transferring component resources,
77 such as processors, is always performed as series of
78 \fBpool_resource_xtransfer()\fR operations, since discrete resources must be
79 identified for transfer.
80 .sp
81 .LP
82 The \fBpool_resource_xtransfer()\fR function transfers the specific
83 \fIcomponents\fR from the \fIsource\fR resource to the \fItarget\fR. Both
84 resources must be of the same type, and of a type that contains components
85 (such as processor sets). The \fIcomponents\fR argument is a null-terminated
86 list of \fBpool_component_t\fR.
87 .sp
88 .LP
89 The \fIconf\fR argument for each function refers to the target configuration to
90 which the operation applies.
91 .SH RETURN VALUES
92 .LP
93 Upon successful completion, \fBpool_resource_create()\fR returns a new
94 \fBpool_resource_t\fR with default properties initialized. Otherwise,
95 \fBINULL\fR is returned and \fBpool_error\fR(3POOL) returns the pool-specific
96 error value.
97 .sp
98 .LP
99 Upon successful completion, \fBpool_resource_destroy()\fR returns 0. Otherwise,
100 -1 is returned and \fBpool_error()\fR returns the pool-specific error value.
101 .sp
102 .LP
103 Upon successful completion, \fBpool_resource_info()\fR returns a string
104 describing the given resource (and optionally its components). Otherwise,
105 \fBINULL\fR is returned and \fBpool_error()\fR returns the pool-specific error
106 value.
107 .sp
108 .LP
109 Upon successful completion, \fBpool_query_resource_components()\fR returns a
110 null-terminated array of \fBpool_component_t * \fR that match the provided
111 null-terminated property list and are contained in the given resource.
112 Otherwise, \fBINULL\fR is returned and \fBpool_error()\fR returns the
113 pool-specific error value.
114 .sp
115 .LP
116 Upon successful completion, \fBpool_resource_transfer()\fR and
117 \fBpool_resource_xtransfer()\fR return 0. Otherwise -1 is returned and
118 \fBpool_error()\fR returns the pool-specific error value.
119 .SH ERRORS
120 .LP
121 Upon successful completion, \fBpool_resource_create()\fR will fail if:
122 .sp

```

```

121 .ne 2
122 .na
123 \fb\fbPOE_BADPARAM\fr\fr
124 .ad
125 .RS 20n
126 The supplied configuration's status is not \fbPOF_VALID\fr or \finame\fr is in
127 use for this resource type.
128 .RE

130 .sp
131 .ne 2
132 .na
133 \fb\fbPOE_INVALID_CONF\fr\fr
134 .ad
135 .RS 20n
136 The resource element could not be created because the configuration would be
137 invalid.
138 .RE

140 .sp
141 .ne 2
142 .na
143 \fb\fbPOE_PUTPROP\fr\fr
144 .ad
145 .RS 20n
146 One of the supplied properties could not be set.
147 .RE

149 .sp
150 .ne 2
151 .na
152 \fb\fbPOE_SYSTEM\fr\fr
153 .ad
154 .RS 20n
155 A system error has occurred. Check the system error code for more details.
156 .RE

158 .sp
159 .LP
160 The \fbpool_resource_destroy()\fr function will fail if:
161 .sp
162 .ne 2
163 .na
164 \fb\fbPOE_BADPARAM\fr\fr
165 .ad
166 .RS 16n
167 The supplied configuration's status is not \fbPOF_VALID\fr.
168 .RE

170 .sp
171 .LP
172 The \fbpool_resource_info()\fr function will fail if:
173 .sp
174 .ne 2
175 .na
176 \fb\fbPOE_BADPARAM\fr\fr
177 .ad
178 .RS 20n
179 The supplied configuration's status is not \fbPOF_VALID\fr or the \fiflags\fr
180 parameter is neither 0 nor 1.
184 parameter is neither 0 nor 1.
181 .RE

183 .sp
184 .ne 2
185 .na

```

```

186 \fb\fbPOE_INVALID_CONF\fr\fr
187 .ad
188 .RS 20n
189 The configuration is invalid.
190 .RE

192 .sp
193 .ne 2
194 .na
195 \fb\fbPOE_SYSTEM\fr\fr
196 .ad
197 .RS 20n
198 A system error has occurred. Check the system error code for more details.
199 .RE

201 .sp
202 .LP
203 The \fbpool_query_resource_components()\fr function will fail if:
204 .sp
205 .ne 2
206 .na
207 \fb\fbPOE_BADPARAM\fr\fr
208 .ad
209 .RS 20n
210 The supplied configuration's status is not \fbPOF_VALID\fr.
211 .RE

213 .sp
214 .ne 2
215 .na
216 \fb\fbPOE_INVALID_CONF\fr\fr
217 .ad
218 .RS 20n
219 The configuration is invalid.
220 .RE

222 .sp
223 .ne 2
224 .na
225 \fb\fbPOE_SYSTEM\fr\fr
226 .ad
227 .RS 20n
228 A system error has occurred. Check the system error code for more details.
229 .RE

231 .sp
232 .LP
233 The \fbpool_resource_transfer()\fr function will fail if:
234 .sp
235 .ne 2
236 .na
237 \fb\fbPOE_BADPARAM\fr\fr
238 .ad
239 .RS 16n
240 The supplied configuration's status is not \fbPOF_VALID\fr, the two resources
241 are not of the same type, or the transfer would cause either of the resources
242 to exceed their \fbmin\fr and \fbmax\fr properties.
243 .RE

245 .sp
246 .ne 2
247 .na
248 \fb\fbPOE_SYSTEM\fr\fr
249 .ad
250 .RS 16n
251 A system error has occurred. Check the system error code for more details.

```

```

252 .RE
254 .sp
255 .LP
256 The \fbpool_resource_xtransfer()\fR function will fail if:
257 .sp
258 .ne 2
259 .na
260 \fb\fbPOE_BADPARAM\fR\fR
261 .ad
262 .RS 20n
263 The supplied configuration's status is not \fbPOF_VALID\fR, the two resources
264 are not of the same type, or the supplied resources do not belong to the
265 source.
266 .RE

268 .sp
269 .ne 2
270 .na
271 \fb\fbPOE_INVALID_CONF\fR\fR
272 .ad
273 .RS 20n
274 The transfer operation failed and the configuration may be invalid.
275 .RE

277 .sp
278 .ne 2
279 .na
280 \fb\fbPOE_SYSTEM\fR\fR
281 .ad
282 .RS 20n
283 A system error has occurred. Check the system error code for more details.
284 .RE

286 .SH EXAMPLES
291 .LP
287 \fbExample 1 \fRCreate a new resource of type \fbpset\fR named \fbfoo\fR.
288 .sp
289 .in +2
290 .nf
291 #include <pool.h>
292 #include <stdio.h>

294 \&...

296 pool_conf_t *conf;
297 pool_resource_t *resource;
298 \&...

300 if ((resource = pool_resource_create(conf, "pset",
301     "foo")) == NULL) {
302     (void) fprintf(stderr, "Cannot create resource\\B{n}");
303     ...
304 }
305 .fi
306 .in -2

308 .SH ATTRIBUTES
314 .LP
309 See \fbAttributes\fR(5) for descriptions of the following attributes:
310 .sp

312 .sp
313 .TS
314 box;
315 c | c

```

```

316 l | l .
317 ATTRIBUTE TYPE ATTRIBUTE VALUE
318 _
319 CSI Enabled
320 _
321 Interface Stability Unstable
322 _
323 MT-Level Safe
324 .TE

326 .SH SEE ALSO
333 .LP
327 \fbBlibpool\fR(3LIB), \fbBpool_error\fR(3POOL), \fbBattributes\fR(5)

```