

new/exception_lists/check_rtime

1

```

*****
10082 Mon Feb 19 17:27:59 2018
new/exception_lists/check_rtime
9124 check_rtime fails with gcc-6-compiled world
Reviewed by: Igor Kozhukhov <igor@dilos.org>
Reviewed by: Toomas Soome <tsoome@me.com>
*****
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 (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 #
23 # Copyright (c) 2009, 2010, Oracle and/or its affiliates. All rights reserved.
24 # Copyright 2015, OmniTI Computer Consulting, Inc. All rights reserved.
25 #
26 #
27 # This file provides exceptions to the usual rules applied to ELF objects by
28 # check_rtime. All strings are Perl regular expressions that are compared to
29 # file paths. In addition to the standard Perl syntax, there is one extension:
30 #
31 #           MACH(dir)
32 #
33 # is expanded into a regular expression that matches the given
34 # directory, or a 64-bit subdirectory of the directory with the
35 # name of a 64-bit architecture. For example, MACH(lib) will match
36 # any of the following:
37 #
38 #           lib
39 #           lib/amd64
40 #           lib/sparcv9
41 #
42 #
43 # Directory hierarchies to skip completely
44 SKIP           ^usr/lib/libc/           # optimized libc
45 SKIP           ^usr/lib/rcm/           # 4426119
46 SKIP           ^usr/perl5/             # alan's taking care of these :-
47 SKIP           ^usr/src/               # no objects in source code
48 #
49 # Individual files that we don't examine
50 SKIP           ^boot/grub/bin/grub$
51 # USIII specific extns. cause ldd noise on USII bld. m/c
52 SKIP           ^usr/lib/fps/sun4u/UltraSPARC.*fpctest$
53 SKIP           ^usr/MACH(lib)/lddstub$ # lddstub has no dependencies
54 SKIP           ^usr/MACH(lib)/libssagent\.so\.$ # 4328854
55 SKIP           ^usr/lib/MACH(iconv)/geniconvtbl.so$ # 4384329
56 #
57 # picl file exclusions (4385799)
58 SKIP           ^usr/platform/.*libpsvcplugin_psr\.so\.$
59 SKIP           ^usr/platform/.*libpsvcpolicy_psr\.so\.$

```

new/exception_lists/check_rtime

2

```

60 SKIP           ^usr/platform/.*libpsvcpolicy\.so\.$
61 SKIP           ^usr/lib/sysevent/modules/picl_slm.so$
62 #
63 # Objects that are allowed to have executable data segments
64 EXEC_DATA      ^MACH(lib)/ld\.so\.$
65 EXEC_DATA      ^lib/libc\.so\.$ # 6524709, 32-bit, needed for x86 only
66 EXEC_DATA      ^lib/amd64/libumem\.so\.$ # ptcumem
67 EXEC_DATA      ^lib/libumem\.so\.$ # ptcumem
68 EXEC_DATA      ^opt/SUNWdtrt/tst/.*ustack/tst\.helper\.exe$
69 EXEC_DATA      ^platform/.*MACH(kernel)/unix$
70 EXEC_DATA      ^platform/.*multiboot$
71 #
72 # Objects that are allowed to have an executable stack
73 EXEC_STACK     ^platform/.*MACH(kernel)/unix$
74 EXEC_STACK     ^platform/.*multiboot$
75 EXEC_STACK     ^opt/os-tests/tests/secflags/stacky$
76 #
77 # Objects for which we allow relocations to the text segment
78 TEXTREL       ^platform/.*MACH(kernel)/unix$
79 #
80 # Directories and files that are allowed to have no direct bound symbols
81 NODIRECT       ^platform/.*MACH(kernel)/unix$
82 NODIRECT       ^usr/ucb
83 NODIRECT       ^usr/4lib/sbcp$
84 #
85 # Identify any files that should be skipped when building a crle(1)
86 # configuration file. As the hwcap libraries can be loop-back mounted onto
87 # libc, these can confuse crle(1) because of their identical dev/inode.
88 NOCRLEALT     ^usr/lib/libc/libc_hwcap[1-3].so.$
89 #
90 # Files that should contain debugging information.
91 STAB          ^platform/.*MACH(kernel)/unix$
92 #
93 # Files that are allowed undefined references
94 UNDEF_REF     ^usr/lib/libnisdb\.so\.$
95 #
96 # Objects allowed to have unused dependencies
97 UNUSED_DEPS   ^usr/lib/picl/plugins/           # require devtree dependencies
98 #
99 # libm.so.2 dependency
100 UNUSED_OBJ   unused object=.*MACH(libm)/libm_hwcap1\.so\.$
101 #
102 # libnetsnmphelpers.so is empty in some net-snmp versions
103 UNUSED_OBJ   unused object=.*libnetsnmphelpers\.so\.$
104 UNREF_OBJ    unreferenced object=.*libnetsnmphelpers\.so\.$
105 #
106 # Unused runpaths due to dlopen() use
107 UNUSED_RPATH /usr/lib/fs/autofs.*\ from\ .automountd
108 UNUSED_RPATH /etc/ppp/plugins.*\ from\ .pppd
109 UNUSED_RPATH /usr/lib/inet/ppp.*\ from\ .pppd
110 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libipsecutil\.so\.$
111 UNUSED_RPATH /usr/platform/.*rsmlib.*\ from\ .*librsm\.so\.$
112 UNUSED_RPATH \${ORIGIN}.*\ from\ .*fcode.so
113 UNUSED_RPATH /opt/VRTSvxvm/lib.*\ from\ .*libdiskmgt\.so\.$
114 #
115 # Unused runpaths in picl code
116 UNUSED_RPATH /usr/platform/.*\ from\ .*/usr/platform
117 UNUSED_RPATH /usr/lib/picl/.*\ from\ .*/usr/platform
118 UNUSED_RPATH /usr/platform/.*\ from\ .*/usr/lib/picl
119 #
120 # Unused runpaths in non-OSNET objects we can't change
121 UNUSED_RPATH /usr/lib/mps.*\ from\ .*libnss3\.so
122 UNUSED_RPATH /usr/lib/mps.*\ from\ .*libnssutil3\.so
123 UNUSED_RPATH /usr/lib/mps.*\ from\ .*libsmime3\.so
124 UNUSED_RPATH /usr/lib/mps.*\ from\ .*libssl3\.so
125 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libdbus-1\.so\.$

```

new/exception_lists/check_rtime

3

```

126 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libdbus-glib-1\.\so\2
127 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libglib-2\.\0\.\so\0
128 UNUSED_RPATH /usr/X11/lib.*\ from\ .*libglib-2\.\0\.\so\0
129 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libgobject-2\.\0\.\so\0
130 UNUSED_RPATH /usr/X11/lib.*\ from\ .*libgobject-2\.\0\.\so\0
131 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libgthread-2\.\0\.\so\0
132 UNUSED_RPATH /usr/X11/lib.*\ from\ .*libgthread-2\.\0\.\so\0
133 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libcrypto\.\so\0\9\8
134 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libnetsnmp\.\so\.*
135 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libgcc_s\.\so\1
136 UNUSED_RPATH /usr/ccs/lib.*\ from\ .*libgcc_s\.\so\1
137 UNUSED_RPATH /usr/lib.*\ from\ .*libgcc_s\.\so\1
138 UNUSED_RPATH /usr/postgres/8.3/lib.*\ from\ .*libpq\.\so\5
139 UNUSED_RPATH /usr/sfw/lib.*\ from\ .*libpq\.\so\5
140 UNUSED_RPATH /usr/lib.*\ from\ .*usr/lib/mps
141 UNUSED_RPATH /usr/ccs/lib.*\ from\ .*usr/lib/mps
142 UNUSED_RPATH /usr/gnu/lib.*\ from\ .*usr/lib/libpython2\.\
143 UNUSED_RPATH /usr/gnu/lib.*\ from\ .*usr/lib/64/libpython2\.\
144 UNUSED_RPATH /usr/snadm/lib.*\ from\ .*usr/snadm/lib/libspmcommon\.\so\1
145 UNUSED_RPATH /usr/gcc/.*\lib.*\ from\ .*
146 #endif /* !codereview */

```

```

149 # Unused runpaths for reasons not captured above
150 UNUSED_RPATH /usr/lib/smb\.*\ from\ .*libsmb\.\so\1 # future needs
151 UNUSED_RPATH /usr.*\ from\ .*tst\.\gcc\.\exe # gcc built

```

```

154 # Unreferenced objects of non-OSnet objects we can't change

```

```

155 UNREF_OBJ /lib.*\ of\ .*libcimapi\.\so
156 UNREF_OBJ /lib.*\ of\ .*libdbus-1\.\so\3
157 UNREF_OBJ /lib.*\ of\ .*libdbus-glib-1\.\so\2
158 UNREF_OBJ /lib.*\ of\ .*libgio-2\.\0\.\so\0
159 UNREF_OBJ /lib.*\ of\ .*libglib-2\.\0\.\so\0
160 UNREF_OBJ /lib.*\ of\ .*libgobject-2\.\0\.\so\0
161 UNREF_OBJ /lib.*\ of\ .*libgthread-2\.\0\.\so\0
162 UNREF_OBJ /lib.*\ of\ .*libjvm\.\so
163 UNREF_OBJ /lib.*\ of\ .*libnetsnmp\.\so\.*
164 UNREF_OBJ /lib.*\ of\ .*libnetsnmpagent\.\so\.*
165 UNREF_OBJ /lib.*\ of\ .*libnetsnmpmibs\.\so\.*
166 UNREF_OBJ /lib.*\ of\ .*libnetsnmphelpers\.\so\.*
167 UNREF_OBJ /lib.*\ of\ .*libspr4\.\so
168 UNREF_OBJ /lib.*\ of\ .*libpq\.\so\5
169 UNREF_OBJ /lib.*\ of\ .*libsoftokn3\.\so
170 UNREF_OBJ /lib.*\ of\ .*libspmcommon\.\so\1
171 UNREF_OBJ /lib.*\ of\ .*libspmcommon\.\so\1
172 UNREF_OBJ /lib.*\ of\ .*libssl3\.\so
173 UNREF_OBJ /lib.*\ of\ .*libtspi\.\so\1
174 UNREF_OBJ /lib.*\ of\ .*libxml2\.\so\2
175 UNREF_OBJ /lib.*\ of\ .*libxslt\.\so\1
176 UNREF_OBJ /lib.*\ of\ .*libpq\.\so\4
177 UNREF_OBJ /lib.*\ of\ .*libpython2\.\4\.\so\1\0
178 UNREF_OBJ /lib.*\ of\ .*libpython2\.\6\.\so\1\0
179 UNREF_OBJ /lib.*\ of\ .*libpython2\.\7\.\so\1\0
180 UNREF_OBJ /libgcc_s.*\ of\ .*libstdc\+\+\.\so\6
181 UNREF_OBJ /libgcc_s.*\ of\ .*libgmodule-2\.\0\.\so\0

```

```

183 # Unreferenced object of objects we can't change for other reasons

```

```

184 UNREF_OBJ /libmapallocc\.\so\1;\ unused\ dependency\ of # interposer
185 UNREF_OBJ /libstdc\+\+\.\so\6;\ unused\ dependency\ of # gcc build
186 UNREF_OBJ /libgcc_s\.\so\1.*\ of\ .*libstdc\+\+\.\so\6 # omnios gcc mix
187 UNREF_OBJ /libm\.\so\2.*\ of\ .*libstdc\+\+\.\so\6 # gcc build
188 UNREF_OBJ /lib.*\ of\ .*lib/picl/plugins/ # picl
189 UNREF_OBJ /lib.*\ of\ .*kcfid # interposer
190 UNREF_OBJ /libpkcs11\.\so\1; .*\ of\ .*libkmf\.\so\1 # interposed
191 # Referenced by the Studio build, not the GCC build. GCC eliminates the unused

```

new/exception_lists/check_rtime

4

```

192 # statics which have the dependence.
193 UNREF_OBJ /libc\.\so\1.*\ of\ .*kldap\.\so\1

```

```

196 # Objects that used to contain system functionality that has since
197 # migrated to libc. We preserve these libraries as pure filters for
198 # backward compatibility but nothing needs to link to them.
199 OLDDEP libaio\.\so\1 # onnv build 44
200 OLDDEP libdl\.\so\1 # on10 build 49
201 OLDDEP libdoor\.\so\1 # onnv build 12
202 OLDDEP libintl\.\so\1 # on297 build 7
203 OLDDEP libpthread\.\so\1 # on10 build 53
204 OLDDEP librt\.\so\1 # onnv build 44
205 OLDDEP libsched\.\so\1 # on10 build 36
206 OLDDEP libthread\.\so\1 # on10 build 53
207 OLDDEP libw\.\so\1 # on297 build 7

```

```

209 # Files for which we skip checking of duplicate addresses in the
210 # symbol sort sections. Such exceptions should be rare --- most code will
211 # not have duplicate addresses, since it takes assembler or a "#pragma weak"
212 # to do such aliasing in C. C++ is different: The compiler generates aliases
213 # for implementation reasons, and the mangled names used to encode argument
214 # and return value types are difficult to handle well in mapfiles.
215 # Furthermore, the Sun compiler and gcc use different and incompatible
216 # name mangling conventions. Since ON must be buildable by either, we
217 # would have to maintain two sets of mapfiles for each such object.
218 # C++ use is rare in ON, so this is not worth pursuing.
219 #

```

```

220 NOSYMSORT opt/SUNWdtrt/tst/common/pid/tst.weak2.exe # DTrace test
221 NOSYMSORT lib/amd64/libns1\.\so\1 # C++
222 NOSYMSORT lib/sparcv9/libns1\.\so\1 # C++
223 NOSYMSORT lib/sparcv9/libfru\.\so\1 # C++
224 NOSYMSORT usr/lib/lms # C++
225 NOSYMSORT ld\.\so\1 # libc_pic.a use
226 NOSYMSORT lib/libsun_fc\.\so\1 # C++
227 NOSYMSORT lib/amd64/libsun_fc\.\so\1 # C++
228 NOSYMSORT lib/sparcv9/libsun_fc\.\so\1 # C++
229 NOSYMSORT usr/lib/amd64/libfru\.\so\1 # C++

```

```

232 # The libprtdiag_psr.so.1 objects built under usr/src/lib/libprtdiag_psr
233 # are a family, all built using the same makefile, targeted at different
234 # sparc hardware variants. There are a small number of cases where this
235 # one size fits all approach causes an object to be linked against an
236 # unneeded library.
237 UNREF_OBJ lib/(libdevinfo|libcfgadm)\.\so\1; .*\ of\ .*SUNW,Netra-CP2300/1

```