

new/usr/src/tools/env/developer.sh

1

```
*****
7950 Thu Aug 14 14:05:27 2014
new/usr/src/tools/env/developer.sh
5092 env files don't need to define LOCKNAME by default
5091 illumos.sh env file's LOCKNAME definition is busted
*****
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) 1999, 2010, Oracle and/or its affiliates. All rights reserved.
24 #
25 #
26 # Configuration variables for the runtime environment of the nightly
27 # build script and other tools for construction and packaging of releases.
28 # This script is sourced by 'nightly' and 'bldenv' to set up the environment
29 # for the build. This example is suitable for building a developers workspace,
30 # which will contain the resulting packages and archives. It is based off
31 # the onnv release. It sets NIGHTLY_OPTIONS to make nightly do:
32 # check ELF ABI/versioning (-A)
33 # runs 'make check' (-C)
34 # DEBUG and non-DEBUG builds (-D)
35 # runs lint in usr/src (-l plus the LINTDIRS variable)
36 # sends mail on completion (-m and the MAILTO variable)
37 # creates packages for PIT/RE (-p)
38 # checks for changes in ELF runpaths (-r)
39 #
40 NIGHTLY_OPTIONS="-ACDlmpr"; export NIGHTLY_OPTIONS
41 #
42 # This is a variable for the rest of the script - GATE doesn't matter to
43 # nightly itself
44 GATE=onnv-bugfixes; export GATE
45 #
46 # CODEMGR_WS - where is your workspace at (or what should nightly name it)
47 CODEMGR_WS="/builds/$GATE"; export CODEMGR_WS
48 #
49 # PARENT_WS is used to determine the parent of this workspace. This is
50 # for the options that deal with the parent workspace (such as where the
51 # proto area will go).
52 #
53 # If you use this, it must be local (or nfs): nightly cannot copy
54 # over ssh or http.
55 PARENT_WS="/ws/onnv-gate"; export PARENT_WS
56 #
57 # CLONE_WS is the workspace nightly should do a bringover from.
58 CLONE_WS="ssh://anonhg@onnv.sfbay.sun.com//export/onnv-clone"; export CLONE_WS
59 #
60 # CLOSED_CLONE_WS is the workspace from which nightly should acquire
```

new/usr/src/tools/env/developer.sh

2

```
61 # the usr/closed tree.
62 CLOSED_CLONE_WS="${CLONE_WS}/usr/closed"; export CLOSED_CLONE_WS
63 #
64 # The bringover, if any, is done as STAFFER.
65 # Set STAFFER to your own login as gatekeeper or developer
66 # The point is to use group "staff" and avoid referencing the parent
67 # workspace as root.
68 # Some scripts optionally send mail messages to MAILTO.
69 #
70 STAFFER=nobody; export STAFFER
71 MAILTO=$STAFFER; export MAILTO
72 #
73 # The project (see project(4)) under which to run this build. If not
74 # specified, the build is simply run in a new task in the current project.
75 BUILD_PROJECT=; export BUILD_PROJECT
76 #
77 # You should not need to change the next three lines
78 # You should not need to change the next four lines
79 LOCKNAME="`basename $CODEMGR_WS`_nightly.lock"; export LOCKNAME
80 ATLOG="$CODEMGR_WS/log"; export ATLOG
81 LOGFILE="$ATLOG/nightly.log"; export LOGFILE
82 MACH=`uname -p`; export MACH
83 #
84 # When the -A flag is specified, and ELF_DATA_BASELINE_DIR is defined,
85 # the ELF interface description file resulting from the build is compared
86 # to that from the specified directory. This ensures that our object
87 # versioning evolves in a backward compatible manner.
88 #
89 # You should not need to change this unless you wish to use locally cached
90 # baseline files. If you use this, it must be local (or nfs): nightly cannot
91 # copy over ssh or http.
92 #
93 ELF_DATA_BASELINE_DIR="/ws/onnv-gate/usr/src/ELF-data-baseline.$MACH"; export E
94 #
95 # This is usually just needed if the closed tree is missing, or when
96 # building a project gate with the -O (cap oh) flag.
97 # ON_CRYPTO_BINS="$PARENT_WS/packages/$MACH/on-crypto.$MACH.tar.bz2"
98 # export ON_CRYPTO_BINS
99 #
100 # REF_PROTO_LIST - for comparing the list of stuff in your proto area
101 # with. Generally this should be left alone, since you want to see differences
102 # from your parent (the gate).
103 #
104 REF_PROTO_LIST=$PARENT_WS/usr/src/proto_list_${MACH}; export REF_PROTO_LIST
105 #
106 #
107 # build environment variables, including version info for mcs, motd,
108 # motd, uname and boot messages. Mostly you shouldn't change this except
109 # when the release slips (nah) or you move an environment file to a new
110 # release
111 #
112 ROOT="$CODEMGR_WS/proto/root_${MACH}"; export ROOT
113 SRC="$CODEMGR_WS/usr/src"; export SRC
114 VERSION="$GATE"; export VERSION
115 #
116 #
117 # the RELEASE and RELEASE_DATE variables are set in Makefile.master;
118 # there might be special reasons to override them here, but that
119 # should not be the case in general
120 #
121 #
122 # RELEASE="5.10.1"; export RELEASE
123 # RELEASE_DATE="October 2007"; export RELEASE_DATE
124 #
125 # proto area in parent for optionally depositing a copy of headers and
126 # libraries corresponding to the protolibs target
127 # not applicable given the NIGHTLY_OPTIONS
```

```

125 #
126 PARENT_ROOT=$PARENT_WS/proto/root_${MACH}; export PARENT_ROOT
127 PARENT_TOOLS_ROOT=$PARENT_WS/usr/src/tools/proto/root_${MACH}-nd; export PARENT_TO

129 #
130 # Package creation variables. You probably shouldn't change these,
131 # either.
132 #
133 # PKGARCHIVE determines where repositories will be created.
134 #
135 # PKGPUBLISHER* control the publisher settings for those repositories.
136 #
137 PKGARCHIVE="${CODEMGR_WS}/packages/${MACH}/nightly"; export PKGARCHIVE
138 # PKGPUBLISHER_REDIST="on-redis"; export PKGPUBLISHER_REDI
139 # PKGPUBLISHER_NONREDIST="on-extra"; export PKGPUBLISHER_NONR

141 # we want make to do as much as it can, just in case there's more than
142 # one problem.
143 MAKEFLAGS=k; export MAKEFLAGS

145 # Magic variable to prevent the devpro compilers/teamware from sending
146 # mail back to devpro on every use.
147 UT_NO_USAGE_TRACKING="1"; export UT_NO_USAGE_TRACKING

149 # Build tools - don't set these unless you know what you're doing. These
150 # variables allows you to get the compilers and onbld files locally or
151 # through cacheofs. Set BUILD_TOOLS to pull everything from one location.
152 # Alternately, you can set ONBLD_TOOLS to where you keep the contents of
153 # SUNWonbld and SPRO_ROOT to where you keep the compilers.
154 #
155 #BUILD_TOOLS=/opt; export BUILD_TOOLS
156 #ONBLD_TOOLS=/opt/onbld; export ONBLD_TOOLS
157 #SPRO_ROOT=/opt/SUNWspro; export SPRO_ROOT

159 # This goes along with lint - it is a series of the form "A [y|n]" which
160 # means "go to directory A and run 'make lint'" Then mail me (y) the
161 # difference in the lint output. 'y' should only be used if the area you're
162 # linting is actually lint clean or you'll get lots of mail.
163 # You shouldn't need to change this though.
164 #LINTDIRS="$SRC y"; export LINTDIRS

166 #
167 # Reference to IA32 IHV workspace, proto area and packages
168 #
169 #IA32_IHV_WS=/ws/${GATE}-ihv; export IA32_IHV_WS
170 #IA32_IHV_ROOT=$IA32_IHV_WS/proto/root_i386; export IA32_IHV_ROOT
171 #IA32_IHV_PKGS=$IA32_IHV_WS/packages/i386/nightly; export IA32_IHV_PKGS

173 #
174 # Reference to binary-only IA32 IHV packages
175 #
176 #IA32_IHV_BINARY_PKGS=/ws/${GATE}-ihv-bin
177 #export IA32_IHV_BINARY_PKGS

179 # Set this flag to 'n' to disable the automatic validation of the dmake
180 # version in use. The default is to check it.
181 #CHECK_DMAKE=y

183 # Set this flag to 'n' to disable the use of 'checkpaths'. The default,
184 # if the 'N' option is not specified, is to run this test.
185 #CHECK_PATHS=y

187 # Set this flag to 'y' to enable the use of elfsigncmp to validate the
188 # output of elfsign. Doing so requires that 't' be set in NIGHTLY_OPTIONS.
189 # The default is to not verify them.
190 #VERIFY_ELFSIGN=n

```

```

192 # BRINGOVER_FILES is the list of files nightly passes to bringover.
193 # If not set the default is "usr", but it can be used for bringing
194 # over deleted_files or other nifty directories.
195 #BRINGOVER_FILES="usr deleted_files"

197 # POST_NIGHTLY can be any command to be run at the end of nightly. See
198 # nightly(1) for interactions between environment variables and this command.
199 #POST_NIGHTLY=

```

new/usr/src/tools/env/gatekeeper.sh

1

```
*****
8563 Thu Aug 14 14:05:27 2014
new/usr/src/tools/env/gatekeeper.sh
5092 env files don't need to define LOCKNAME by default
5091 illumos.sh env file's LOCKNAME definition is busted
*****
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) 1999, 2010, Oracle and/or its affiliates. All rights reserved.
24 #
25 #
26 # Configuration variables for the runtime environment of the nightly
27 # build script and other tools for construction and packaging of releases.
28 # This script is sourced by 'nightly' and 'blendv' to set up the environment
29 # for the build. This example is suitable for building a gate,
30 # which will contain the resulting packages and archives (builds of the gate
31 # are done in children and then the resulting archives, packages, and proto
32 # area are put into the parent for everyone to use). It is based off
33 # the onnv release. It sets NIGHTLY_OPTIONS to make nightly do:
34 # DEBUG and non-DEBUG builds (-D)
35 # creates packages for PIT/RE (-p)
36 # checks for new interfaces in libraries (-A)
37 # runs 'make check' (-C)
38 # runs lint in usr/src (-l plus the LINTDIRS variable)
39 # sends mail on completion (-m and the MAILTO variable)
40 # updates the protolist in the parent for children to compare with (-u)
41 # updates the proto area in the parent when done (-U)
42 # checks for changes in ELF runpaths (-r)
43 # checks for changes in unreferenced files (-f)
44 #
45 NIGHTLY_OPTIONS="-ADClmpUrf"; export NIGHTLY_OPTIONS
46 #
47 # This is a variable for the rest of the script - GATE doesn't matter to
48 # nightly itself
49 GATE=onnv-gate; export GATE
50 #
51 # CODEMGR_WS - where is your workspace at (or what should nightly name it)
52 # there is only one definition here, which assumes all the gate build machines
53 # (sparc and x86) are set up the same. But remember, this is a script, so
54 # you _could_ look at $MACH or 'uname -n' and set these variables differently.
55 CODEMGR_WS="/builds/$GATE"; export CODEMGR_WS
56 #
57 # PARENT_WS is used to determine the parent of this workspace. This is
58 # for the options that deal with the parent workspace (such as where the
59 # proto area will go).
60 #
```

new/usr/src/tools/env/gatekeeper.sh

2

```
61 # If you use this, it must be local (or nfs): nightly cannot copy
62 # over ssh or http.
63 PARENT_WS="/ws/$GATE"; export PARENT_WS
64 #
65 # CLONE_WS is the workspace nightly should do a bringover from.
66 CLONE_WS="ssh://anonhg@onnv.sfbay.sun.com//export/onnv-clone"; export CLONE_WS
67 #
68 # CLOSED_CLONE_WS is the workspace from which nightly will acquire the
69 # usr/closed tree.
70 CLOSED_CLONE_WS="${CLONE_WS}/usr/closed"
71 export CLOSED_CLONE_WS
72 #
73 # The bringover, if any, is done as STAFFER.
74 # Set STAFFER to your own login as gatekeeper or integration engineer.
75 # The point is to use group "staff" and avoid referencing the parent
76 # workspace as root.
77 # Some scripts optionally send mail messages to MAILTO.
78 #
79 STAFFER=nobody; export STAFFER
80 MAILTO=$STAFFER; export MAILTO
81 #
82 # The project (see project(4)) under which to run this build. If not
83 # specified, the build is simply run in a new task in the current project.
84 BUILD_PROJECT=; export BUILD_PROJECT
85 #
86 # You should not need to change the next three lines
87 # You should not need to change the next four lines
87 LOCKNAME='basename $CODEMGR_WS'_nightly.lock"; export LOCKNAME
87 ATLOG="$CODEMGR_WS/log"; export ATLOG
88 LOGFILE="$ATLOG/nightly.log"; export LOGFILE
89 MACH='uname -p'; export MACH
90 #
91 # When the -A flag is specified, and ELF_DATA_BASELINE_DIR is defined,
92 # the ELF interface description file resulting from the build is compared
93 # to that from the specified directory. This ensures that our object
94 # versioning evolves in a backward compatible manner.
95 #
96 # You should not need to change this unless you wish to use locally cached
97 # baseline files. If you use this, it must be local (or nfs): nightly cannot
98 # copy over ssh or http.
99 #
100 ELF_DATA_BASELINE_DIR="/ws/onnv-gate/usr/src/ELF-data-baseline.$MACH"; export E
101 #
102 # This is usually just needed if the closed tree is missing, or when
103 # building a project gate with the -O (cap oh) flag.
104 # ON_CRYPT_BIN="$PARENT_WS/packages/$MACH/on-crypto.$MACH.tar.bz2"
105 # export ON_CRYPT_BIN
106 #
107 # REF_PROTO_LIST - for comparing the list of stuff in your proto area
108 # with. Generally this should be left alone, since you want to see differences
109 # between today's build and yesterdays.
110 #
111 REF_PROTO_LIST=$PARENT_WS/usr/src/proto_list_{$MACH}; export REF_PROTO_LIST
112 #
113 #
114 # build environment variables, including version info for mcs, motd,
115 # motd, uname and boot messages. Mostly you shouldn't change this except
116 # when the release slips (nah) or when starting a new release.
117 #
118 ROOT="$CODEMGR_WS/proto/root_{$MACH}"; export ROOT
119 SRC="$CODEMGR_WS/usr/src"; export SRC
120 VERSION="$GATE"; export VERSION
121 #
122 #
123 # the RELEASE and RELEASE_DATE variables are set in Makefile.master;
124 # there might be special reasons to override them here, but that
```

new/usr/src/tools/env/gatekeeper.sh

3

```
125 # should not be the case in general
126 #
127 # RELEASE="5.10.1";                export RELEASE
128 # RELEASE_DATE="October 2007";    export RELEASE_DATE

130 # proto area in parent for optionally depositing a copy of headers and
131 # libraries corresponding to the protolibs target
132 #
133 PARENT_ROOT=$PARENT_WS/proto/root_${MACH}; export PARENT_ROOT
134 PARENT_TOOLS_ROOT=$PARENT_WS/usr/src/tools/proto/root_${MACH}-nd; export PARENT_TO

136 #
137 # Package creation variables. You probably shouldn't change these,
138 # either.
139 #
140 # PKGARCHIVE determines where repositories will be created.
141 #
142 # PKGPUBLISHER* control the publisher settings for those repositories.
143 #
144 PKGARCHIVE="${PARENT_WS}/packages/${MACH}/nightly"; export PKGARCHIVE
145 # PKGPUBLISHER_REDIST="on-nightly"; export PKGPUBLISHER_REDIST
146 # PKGPUBLISHER_NONREDIST="on-extra"; export PKGPUBLISHER_NONREDIST

149 # we want make to do as much as it can, just in case there's more than
150 # one problem. This is especially important with the gate, since multiple
151 # unrelated broken things can be integrated.
152 MAKEFLAGS=k; export MAKEFLAGS

154 # Magic variable to prevent the devpro compilers/teamware from sending
155 # mail back to devpro on every use.
156 UT_NO_USAGE_TRACKING="1"; export UT_NO_USAGE_TRACKING

158 # Build tools - don't set these unless you know what you're doing. These
159 # variables allows you to get the compilers and onbld files locally or
160 # through cachefs. Set BUILD_TOOLS to pull everything from one location.
161 # Alternately, you can set ONBLD_TOOLS to where you keep the contents of
162 # SUNWwonbld and SPRO_ROOT to where you keep the compilers.
163 #
164 #BUILD_TOOLS=/opt;                export BUILD_TOOLS
165 #ONBLD_TOOLS=/opt/onbld;          export ONBLD_TOOLS
166 #SPRO_ROOT=/opt/SUNspro;          export SPRO_ROOT

168 # This goes along with lint - it is a series of the form "A [y|n]" which
169 # means "go to directory A and run 'make lint'" Then mail me (y) the
170 # difference in the lint output. 'y' should only be used if the area you're
171 # linting is actually lint clean or you'll get lots of mail.
172 # You shouldn't need to change this though.
173 #LINTDIRS="$SRC y"; export LINTDIRS

175 #
176 # Reference to IA32 IHV workspace, proto area and packages
177 #
178 #IA32_IHV_WS=/ws/${GATE}-ihv;     export IA32_IHV_WS
179 #IA32_IHV_ROOT=$IA32_IHV_WS/proto/root_i386; export IA32_IHV_ROOT
180 #IA32_IHV_PKGS=$IA32_IHV_WS/packages/i386/nightly; export IA32_IHV_PKGS

182 #
183 # Reference to binary-only IA32 IHV packages
184 #
185 #IA32_IHV_BINARY_PKGS=/ws/${GATE}-ihv-bin
186 #export IA32_IHV_BINARY_PKGS

188 # Set this flag to 'n' to disable the automatic validation of the dmake
189 # version in use. The default is to check it.
190 #CHECK_DMAKE=y
```

new/usr/src/tools/env/gatekeeper.sh

4

```
192 # Set this flag to 'n' to disable the use of 'checkpaths'. The default,
193 # if the 'N' option is not specified, is to run this test.
194 #CHECK_PATHS=y

196 # Set this flag to 'y' to enable the use of elfsigncmp to validate the
197 # output of elfsign. Doing so requires that 't' be set in NIGHTLY_OPTIONS.
198 # The default is to not verify them.
199 #VERIFY_ELFSIGN=n

201 # BRINGOVER_FILES is the list of files nightly passes to bringover.
202 # If not set the default is "usr", but it can be used for bringing
203 # over deleted_files or other nifty directories.
204 #BRINGOVER_FILES="usr deleted_files"

206 # POST_NIGHTLY can be any command to be run at the end of nightly. See
207 # nightly(1) for interactions between environment variables and this command.
208 #POST_NIGHTLY=
```

new/usr/src/tools/env/illumos.sh

1

```
*****
      8410 Thu Aug 14 14:05:27 2014
new/usr/src/tools/env/illumos.sh
5092 env files don't need to define LOCKNAME by default
5091 illumos.sh env file's LOCKNAME definition is busted
*****
_____
unchanged_portion_omitted_
```

```
100 maxjobs DMAKE_MAX_JOBS # "DMAKE_MAX_JOBS" passed as ksh(1) name reference
101 export DMAKE_MAX_JOBS
```

```
103 # path to onbld tool binaries
104 ONBLD_BIN='/opt/onbld/bin'
```

```
106 # PARENT_WS is used to determine the parent of this workspace. This is
107 # for the options that deal with the parent workspace (such as where the
108 # proto area will go).
109 export PARENT_WS=''
```

```
111 # CLONE_WS is the workspace nightly should do a bringover from.
112 export CLONE_WS='ssh://anonhg@hg.illumos.org/illumos-gate'
```

```
114 # The bringover, if any, is done as STAFFER.
115 # Set STAFFER to your own login as gatekeeper or developer
116 # The point is to use group "staff" and avoid referencing the parent
117 # workspace as root.
118 # Some scripts optionally send mail messages to MAILTO.
119 #
120 export STAFFER="$LOGNAME"
121 export MAILTO="$STAFFER"
```

```
123 # If you wish the mail messages to be From: an arbitrary address, export
124 # MAILFROM.
125 #export MAILFROM="user@example.com"
```

```
127 # The project (see project(4)) under which to run this build. If not
128 # specified, the build is simply run in a new task in the current project.
129 export BUILD_PROJECT=''
```

```
131 # You should not need to change the next three lines
131 # You should not need to change the next four lines
132 export LOCKNAME="${basename -- "$CODEMGR_WS"}_nightly.lock"
132 export ATLOG="$CODEMGR_WS/log"
133 export LOGFILE="$ATLOG/nightly.log"
134 export MACH="${uname -p}"
```

```
136 #
137 # The following two macros are the closed/crypto binaries. Once
138 # Illumos has totally freed itself, we can remove these references.
139 #
140 # Location of encumbered binaries.
141 export ON_CLOSED_BINS="$CODEMGR_WS/closed"
142 # Location of signed cryptographic binaries.
143 export ON_CRYPTO_BINS="$CODEMGR_WS/on-crypto.$MACH.tar.bz2"
```

```
145 # REF_PROTO_LIST - for comparing the list of stuff in your proto area
146 # with. Generally this should be left alone, since you want to see differences
147 # from your parent (the gate).
148 #
149 export REF_PROTO_LIST="$PARENT_WS/usr/src/proto_list_${MACH}"
```

```
152 export ROOT="$CODEMGR_WS/proto/root_${MACH}"
153 export SRC="$CODEMGR_WS/usr/src"
154 export MULTI_PROTO="no"
```

new/usr/src/tools/env/illumos.sh

2

```
156 #
157 # build environment variables, including version info for mcs, motd,
158 # motd, uname and boot messages. Mostly you shouldn't change this except
159 # when the release slips (mah) or you move an environment file to a new
160 # release
161 #
162 export VERSION="$GATE"
```

```
164 #
165 # the RELEASE and RELEASE_DATE variables are set in Makefile.master;
166 # there might be special reasons to override them here, but that
167 # should not be the case in general
168 #
169 # export RELEASE='5.11'
170 # export RELEASE_DATE='October 2007'
```

```
172 # proto area in parent for optionally depositing a copy of headers and
173 # libraries corresponding to the protolib target
174 # not applicable given the NIGHTLY_OPTIONS
175 #
176 export PARENT_ROOT="$PARENT_WS/proto/root_${MACH}"
177 export PARENT_TOOLS_ROOT="$PARENT_WS/usr/src/tools/proto/root_${MACH}-nd"
```

```
179 # Package creation variables. You probably shouldn't change these,
180 # either.
181 #
182 # PKGARCHIVE determines where the repository will be created.
183 #
184 # PKGPUBLISHER_REDIST controls the publisher setting for the repository.
185 #
186 export PKGARCHIVE="${CODEMGR_WS}/packages/${MACH}/nightly"
187 # export PKGPUBLISHER_REDIST='on-redist'
```

```
189 # Package manifest format version.
190 export PKGFMT_OUTPUT='v1'
```

```
192 # we want make to do as much as it can, just in case there's more than
193 # one problem.
194 export MAKEFLAGS='k'
```

```
196 # Magic variable to prevent the devpro compilers/teamware from sending
197 # mail back to devpro on every use.
198 export UT_NO_USAGE_TRACKING='1'
```

```
200 # Build tools - don't change these unless you know what you're doing. These
201 # variables allows you to get the compilers and onbld files locally or
202 # through cacheofs. Set BUILD_TOOLS to pull everything from one location.
203 # Alternately, you can set ONBLD_TOOLS to where you keep the contents of
204 # SUNWonbld and SPRO_ROOT to where you keep the compilers. SPRO_VROOT
205 # exists to make it easier to test new versions of the compiler.
206 export BUILD_TOOLS='/opt'
207 #export ONBLD_TOOLS='/opt/onbld'
208 export SPRO_ROOT='/opt/SUNWspro'
209 export SPRO_VROOT="$SPRO_ROOT"
```

```
211 # This goes along with lint - it is a series of the form "A [y|n]" which
212 # means "go to directory A and run 'make lint'" Then mail me (y) the
213 # difference in the lint output. 'y' should only be used if the area you're
214 # linting is actually lint clean or you'll get lots of mail.
215 # You shouldn't need to change this though.
216 #export LINTDIRS="$SRC y"
```

```
218 # Set this flag to 'n' to disable the automatic validation of the dmake
219 # version in use. The default is to check it.
220 #CHECK_DMAKE='y'
```

**new/usr/src/tools/env/illumos.sh**

**3**

```
222 # Set this flag to 'n' to disable the use of 'checkpaths'. The default,
223 # if the 'N' option is not specified, is to run this test.
224 #CHECK_PATHS='y'

226 # POST_NIGHTLY can be any command to be run at the end of nightly. See
227 # nightly(1) for interactions between environment variables and this command.
228 #POST_NIGHTLY=

230 # Uncomment this to disable support for SMB printing.
231 # export ENABLE_SMB_PRINTING='#'
```