

new/usr/src/tools/scripts/nightly.sh

1

```
*****
86599 Wed Jul 30 20:44:01 2008
new/usr/src/tools/scripts/nightly.sh
6730623 nightly should create usr/closed if explicitly asked
6731128 nightly + mercurial bringover messaging could be improved
*****
1 #!/bin/ksh -p
2 #
3 # CDDL HEADER START
4 #
5 # The contents of this file are subject to the terms of the
6 # Common Development and Distribution License (the "License").
7 # You may not use this file except in compliance 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 #
24 # Copyright 2008 Sun Microsystems, Inc. All rights reserved.
25 # Use is subject to license terms.
26 #
27 # ident "@(#)nightly.sh 1.124 08/07/30 SMI"
27 # ident "@(#)nightly.sh 1.123 08/07/16 SMI"
28 #
29 # Based on the nightly script from the integration folks,
30 # Mostly modified and owned by mike_s.
31 # Changes also by kjc, dmK.
32 #
33 # BRINGOVER_WS may be specified in the env file.
34 # The default is the old behavior of CLONE_WS
35 #
36 # -i on the command line, means fast options, so when it's on the
37 # command line (only), lint and check builds are skipped no matter what
38 # the setting of their individual flags are in NIGHTLY_OPTIONS.
39 #
40 # LINTDIRS can be set in the env file, format is a list of:
41 #
42 # /dirname-to-run-lint-on flag
43 #
44 # Where flag is: y - enable lint noise diff output
45 # n - disable lint noise diff output
46 #
47 # For example: LINTDIRS="$SRC/uts n $SRC/stand y $SRC/psm y"
48 #
49 # -A flag in NIGHTLY_OPTIONS checks ABI diffs in .so files
50 # This option requires a couple of scripts.
51 #
52 # OPTHOME and TEAMWARE may be set in the environment to override /opt
53 # and /opt/teamware defaults.
54 #
55 #
56 #
57 # The CDPATH variable causes ksh's 'cd' builtin to emit messages to stdout
58 # under certain circumstances, which can really screw things up; unset it.
59 #
```

new/usr/src/tools/scripts/nightly.sh

2

```
60 unset CDPATH
62 # Get the absolute path of the nightly script that the user invoked. This
63 # may be a relative path, and we need to do this before changing directory.
64 nightly_path='whence $0'
65 #
66 #
67 # Keep track of where we found nightly so we can invoke the matching
68 # which_scm script. If that doesn't work, don't go guessing, just rely
69 # on the $PATH settings, which will generally give us either /opt/onbld
70 # or the user's workspace.
71 #
72 WHICH_SCM=$(dirname $nightly_path)/which_scm
73 if [[ ! -x $WHICH_SCM ]]; then
74     WHICH_SCM=which_scm
75 fi
76 #
77 #
78 # Print the tag string used to identify a build (e.g., "DEBUG
79 # open-only")
80 # usage: tagstring debug-part open-part
81 #
82 tagstring() {
83     debug_part=$1
84     open_part=$2
85 #
86     if [ -n "$open_part" ]; then
87         echo "$debug_part $open_part"
88     else
89         echo "$debug_part"
90     fi
91 }
92 #
93 # _____unchanged_portion_omitted_____
94 #
95 #
96 #
97 #
98 # Verify that the closed tree is present if it needs to be.
99 # Sets CLOSED_IS_PRESENT for future use.
100 #
101 #
102 check_closed_tree() {
103     if [ -z "$CLOSED_IS_PRESENT" ]; then
104         if [ -d $CODEMGR_WS/usr/closed ]; then
105             if [ -d $SRC/./closed ]; then
106                 CLOSED_IS_PRESENT="yes"
107             else
108                 CLOSED_IS_PRESENT="no"
109             fi
110         fi
111         export CLOSED_IS_PRESENT
112     fi
113     if [[ "$CLOSED_IS_PRESENT" = no && ! -d "$ON_CLOSED_BINS" ]]; then
114         #
115         # If it's an old (pre-split) tree or an empty
116         # workspace, don't complain.
117         #
118         if grep -s CLOSED_BUILD $SRC/Makefile.master > /dev/null; then
119             echo "If the closed sources are not present, \" \
120                 \"ON_CLOSED_BINS"
121             echo "must point to the closed binaries tree."
122             build_ok=n
123             exit 1
124         fi
125     fi
126 }
127 #
128 # _____unchanged_portion_omitted_____
129 #
130 #
131 #
132 #
133 #
134 #
135 #
136 #
137 #
138 #
139 #
140 #
141 #
142 #
143 #
144 #
145 #
146 #
147 #
148 #
149 #
150 #
151 #
152 #
153 #
154 #
155 #
156 #
157 #
158 #
159 #
160 #
161 #
162 #
163 #
164 #
165 #
166 #
167 #
168 #
169 #
170 #
171 #
172 #
173 #
174 #
175 #
176 #
177 #
178 #
179 #
180 #
181 #
182 #
183 #
184 #
185 #
186 #
187 #
188 #
189 #
190 #
191 #
192 #
193 #
194 #
195 #
196 #
197 #
198 #
199 #
200 #
201 #
202 #
203 #
204 #
205 #
206 #
207 #
208 #
209 #
210 #
211 #
212 #
213 #
214 #
215 #
216 #
217 #
218 #
219 #
220 #
221 #
222 #
223 #
224 #
225 #
226 #
227 #
228 #
229 #
230 #
231 #
232 #
233 #
234 #
235 #
236 #
237 #
238 #
239 #
240 #
241 #
242 #
243 #
244 #
245 #
246 #
247 #
248 #
249 #
250 #
251 #
252 #
253 #
254 #
255 #
256 #
257 #
258 #
259 #
260 #
261 #
262 #
263 #
264 #
265 #
266 #
267 #
268 #
269 #
270 #
271 #
272 #
273 #
274 #
275 #
276 #
277 #
278 #
279 #
280 #
281 #
282 #
283 #
284 #
285 #
286 #
287 #
288 #
289 #
290 #
291 #
292 #
293 #
294 #
295 #
296 #
297 #
298 #
299 #
300 #
301 #
302 #
303 #
304 #
305 #
306 #
307 #
308 #
309 #
310 #
311 #
312 #
313 #
314 #
315 #
316 #
317 #
318 #
319 #
320 #
321 #
322 #
323 #
324 #
325 #
326 #
327 #
328 #
329 #
330 #
331 #
332 #
333 #
334 #
335 #
336 #
337 #
338 #
339 #
340 #
341 #
342 #
343 #
344 #
345 #
346 #
347 #
348 #
349 #
350 #
351 #
352 #
353 #
354 #
355 #
356 #
357 #
358 #
359 #
360 #
361 #
362 #
363 #
364 #
365 #
366 #
367 #
368 #
369 #
370 #
371 #
372 #
373 #
374 #
375 #
376 #
377 #
378 #
379 #
380 #
381 #
382 #
383 #
384 #
385 #
386 #
387 #
388 #
389 #
390 #
391 #
392 #
393 #
394 #
395 #
396 #
397 #
398 #
399 #
400 #
401 #
402 #
403 #
404 #
405 #
406 #
407 #
408 #
409 #
410 #
411 #
412 #
413 #
414 #
415 #
416 #
417 #
418 #
419 #
420 #
421 #
422 #
423 #
424 #
425 #
426 #
427 #
428 #
429 #
430 #
431 #
432 #
433 #
434 #
435 #
436 #
437 #
438 #
439 #
440 #
441 #
442 #
443 #
444 #
445 #
446 #
447 #
448 #
449 #
450 #
451 #
452 #
453 #
454 #
455 #
456 #
457 #
458 #
459 #
460 #
461 #
462 #
463 #
464 #
465 #
466 #
467 #
468 #
469 #
470 #
471 #
472 #
473 #
474 #
475 #
476 #
477 #
478 #
479 #
480 #
481 #
482 #
483 #
484 #
485 #
486 #
487 #
488 #
489 #
490 #
491 #
492 #
493 #
494 #
495 #
496 #
497 #
498 #
499 #
500 #
501 #
502 #
503 #
504 #
505 #
506 #
507 #
508 #
509 #
510 #
511 #
512 #
513 #
514 #
515 #
516 #
517 #
518 #
519 #
520 #
521 #
522 #
523 #
524 #
525 #
526 #
527 #
528 #
529 #
530 #
531 #
532 #
533 #
534 #
535 #
536 #
537 #
538 #
539 #
540 #
541 #
542 #
543 #
544 #
545 #
546 #
547 #
548 #
549 #
550 #
551 #
552 #
553 #
554 #
555 #
556 #
557 #
558 #
559 #
560 #
561 #
562 #
563 #
564 #
565 #
566 #
567 #
568 #
569 #
570 #
571 #
572 #
573 #
574 #
575 #
576 #
577 #
578 #
579 #
580 #
581 #
582 #
583 #
584 #
585 #
586 #
587 #
588 #
589 #
590 #
591 #
592 #
593 #
594 #
595 #
596 #
597 #
598 #
599 #
600 #
601 #
602 #
603 #
604 #
605 #
606 #
607 #
608 #
609 #
610 #
611 #
612 #
613 #
614 #
615 #
616 #
617 #
618 #
619 #
620 #
621 #
622 #
623 #
624 #
625 #
626 #
627 #
628 #
629 #
630 #
631 #
632 #
633 #
634 #
635 #
636 #
637 #
638 #
639 #
640 #
641 #
642 #
643 #
644 #
645 #
646 #
647 #
648 #
649 #
650 #
651 #
652 #
653 #
654 #
655 #
656 #
657 #
658 #
659 #
660 #
661 #
662 #
663 #
664 #
665 #
666 #
667 #
668 #
669 #
670 #
671 #
672 #
673 #
674 #
675 #
676 #
677 #
678 #
679 #
680 #
681 #
682 #
683 #
684 #
685 #
686 #
687 #
688 #
689 #
690 #
691 #
692 #
693 #
694 #
695 #
696 #
697 #
698 #
699 #
700 #
701 #
702 #
703 #
704 #
705 #
706 #
707 #
708 #
709 #
710 #
711 #
712 #
713 #
714 #
715 #
716 #
717 #
718 #
719 #
720 #
721 #
722 #
723 #
724 #
725 #
726 #
727 #
728 #
729 #
730 #
731 #
732 #
733 #
734 #
735 #
736 #
737 #
738 #
739 #
740 #
741 #
742 #
743 #
744 #
745 #
746 #
747 #
748 #
749 #
750 #
751 #
752 #
753 #
754 #
755 #
756 #
757 #
758 #
759 #
760 #
761 #
762 #
763 #
764 #
765 #
766 #
767 #
768 #
769 #
770 #
771 #
772 #
773 #
774 #
775 #
776 #
777 #
778 #
779 #
780 #
781 #
782 #
783 #
784 #
785 #
786 #
787 #
788 #
789 #
790 #
791 #
792 #
793 #
794 #
795 #
796 #
797 #
798 #
799 #
800 #
801 #
802 #
803 #
804 #
805 #
806 #
807 #
808 #
809 #
810 #
811 #
812 #
813 #
814 #
815 #
816 #
817 #
818 #
819 #
820 #
821 #
822 #
823 #
824 #
825 #
826 #
827 #
828 #
829 #
830 #
831 #
832 #
833 #
834 #
835 #
836 #
837 #
838 #
839 #
840 #
841 #
842 #
843 #
844 #
845 #
846 #
847 #
848 #
849 #
850 #
851 #
852 #
853 #
854 #
855 #
856 #
857 #
858 #
859 #
860 #
861 #
862 #
863 #
864 #
865 #
866 #
867 #
868 #
869 #
870 #
871 #
872 #
873 #
874 #
875 #
876 #
877 #
878 #
879 #
880 #
881 #
882 #
883 #
884 #
885 #
886 #
887 #
888 #
889 #
890 #
891 #
892 #
893 #
894 #
895 #
896 #
897 #
898 #
899 #
900 #
901 #
902 #
903 #
904 #
905 #
906 #
907 #
908 #
909 #
910 #
911 #
912 #
913 #
914 #
915 #
916 #
917 #
918 #
919 #
920 #
921 #
922 #
923 #
924 #
925 #
926 #
927 #
928 #
929 #
930 #
931 #
932 #
933 #
934 #
935 #
936 #
937 #
938 #
939 #
940 #
941 #
942 #
943 #
944 #
945 #
946 #
947 #
948 #
949 #
950 #
951 #
952 #
953 #
954 #
955 #
956 #
957 #
958 #
959 #
960 #
961 #
962 #
963 #
964 #
965 #
966 #
967 #
968 #
969 #
970 #
971 #
972 #
973 #
974 #
975 #
976 #
977 #
978 #
979 #
980 #
981 #
982 #
983 #
984 #
985 #
986 #
987 #
988 #
989 #
990 #
991 #
992 #
993 #
994 #
995 #
996 #
997 #
998 #
999 #
1000 #
1001 #
1002 #
1003 #
1004 #
1005 #
1006 #
1007 #
1008 #
1009 #
1010 #
1011 #
1012 #
1013 #
1014 #
1015 #
1016 #
1017 #
1018 #
1019 #
1020 #
1021 #
1022 #
1023 #
1024 #
1025 #
1026 #
1027 #
1028 #
1029 #
1030 #
1031 #
1032 #
1033 #
1034 #
1035 #
1036 #
1037 #
1038 #
1039 #
1040 #
1041 #
1042 #
1043 #
1044 #
1045 #
1046 #
1047 #
1048 #
1049 #
1050 #
1051 #
1052 #
1053 #
1054 #
1055 #
1056 #
1057 #
1058 #
1059 #
1060 #
1061 #
1062 #
1063 #
1064 #
1065 #
1066 #
1067 #
1068 #
1069 #
1070 #
1071 #
1072 #
1073 #
1074 #
1075 #
1076 #
1077 #
1078 #
1079 #
1080 #
1081 #
1082 #
1083 #
1084 #
1085 #
1086 #
1087 #
1088 #
1089 #
1090 #
1091 #
1092 #
1093 #
1094 #
1095 #
1096 #
1097 #
1098 #
1099 #
1100 #
1101 #
1102 #
1103 #
1104 #
1105 #
1106 #
1107 #
1108 #
1109 #
1110 #
1111 #
1112 #
1113 #
1114 #
1115 #
1116 #
1117 #
1118 #
1119 #
1120 #
1121 #
1122 #
1123 #
1124 #
1125 #
1126 #
1127 #
1128 #
1129 #
1130 #
1131 #
1132 #
1133 #
1134 #
1135 #
1136 #
1137 #
1138 #
1139 #
1140 #
1141 #
1142 #
1143 #
1144 #
1145 #
1146 #
1147 #
1148 #
1149 #
1150 #
1151 #
1152 #
1153 #
1154 #
1155 #
1156 #
1157 #
1158 #
1159 #
1160 #
1161 #
1162 #
1163 #
1164 #
1165 #
1166 #
1167 #
1168 #
1169 #
1170 #
1171 #
1172 #
1173 #
1174 #
1175 #
1176 #
1177 #
1178 #
1179 #
1180 #
1181 #
1182 #
1183 #
1184 #
1185 #
1186 #
1187 #
1188 #
1189 #
1190 #
1191 #
1192 #
1193 #
1194 #
1195 #
1196 #
1197 #
1198 #
1199 #
1200 #
1201 #
1202 #
1203 #
1204 #
1205 #
1206 #
1207 #
1208 #
1209 #
1210 #
1211 #
1212 #
1213 #
1214 #
1215 #
1216 #
1217 #
1218 #
1219 #
1220 #
1221 #
1222 #
1223 #
1224 #
1225 #
1226 #
1227 #
1228 #
1229 #
1230 #
1231 #
1232 #
1233 #
1234 #
1235 #
1236 #
1237 #
1238 #
1239 #
1240 #
1241 #
1242 #
1243 #
1244 #
1245 #
1246 #
1247 #
1248 #
1249 #
1250 #
1251 #
1252 #
1253 #
1254 #
1255 #
1256 #
1257 #
1258 #
1259 #
1260 #
1261 #
1262 #
1263 #
1264 #
1265 #
1266 #
1267 #
1268 #
1269 #
1270 #
1271 #
1272 #
1273 #
1274 #
1275 #
1276 #
1277 #
1278 #
1279 #
1280 #
1281 #
1282 #
1283 #
1284 #
1285 #
1286 #
1287 #
1288 #
1289 #
1290 #
1291 #
1292 #
1293 #
1294 #
1295 #
1296 #
1297 #
1298 #
1299 #
1300 #
1301 #
1302 #
1303 #
1304 #
1305 #
1306 #
1307 #
1308 #
1309 #
1310 #
1311 #
1312 #
1313 #
1314 #
1315 #
1316 #
1317 #
1318 #
1319 #
1320 #
1321 #
1322 #
1323 #
1324 #
1325 #
1326 #
1327 #
1328 #
1329 #
1330 #
1331 #
1332 #
1333 #
1334 #
1335 #
1336 #
1337 #
1338 #
1339 #
1340 #
1341 #
1342 #
1343 #
1344 #
1345 #
1346 #
1347 #
1348 #
1349 #
1350 #
1351 #
1352 #
1353 #
1354 #
1355 #
1356 #
1357 #
1358 #
1359 #
1360 #
1361 #
1362 #
1363 #
1364 #
1365 #
1366 #
1367 #
1368 #
1369 #
1370 #
1371 #
1372 #
1373 #
1374 #
1375 #
1376 #
1377 #
1378 #
1379 #
1380 #
1381 #
1382 #
1383 #
1384 #
1385 #
1386 #
1387 #
1388 #
1389 #
1390 #
1391 #
1392 #
1393 #
1394 #
1395 #
1396 #
1397 #
1398 #
1399 #
1400 #
1401 #
1402 #
1403 #
1404 #
1405 #
1406 #
1407 #
1408 #
1409 #
1410 #
1411 #
1412 #
1413 #
1414 #
1415 #
1416 #
1417 #
1418 #
1419 #
1420 #
1421 #
1422 #
1423 #
1424 #
1425 #
1426 #
1427 #
1428 #
1429 #
1430 #
1431 #
1432 #
1433 #
1434 #
1435 #
1436 #
1437 #
1438 #
1439 #
1440 #
1441 #
1442 #
1443 #
1444 #
1445 #
1446 #
1447 #
1448 #
1449 #
1450 #
1451 #
1452 #
1453 #
1454 #
1455 #
1456 #
1457 #
1458 #
1459 #
1460 #
1461 #
1462 #
1463 #
1464 #
1465 #
1466 #
1467 #
1468 #
1469 #
1470 #
1471 #
1472 #
1473 #
1474 #
1475 #
1476 #
1477 #
1478 #
1479 #
1480 #
1481 #
1482 #
1483 #
1484 #
1485 #
1486 #
1487 #
1488 #
1489 #
1490 #
1491 #
1492 #
1493 #
1494 #
1495 #
1496 #
1497 #
1498 #
1499 #
1500 #
1501 #
1502 #
1503 #
1504 #
1505 #
1506 #
1507 #
1508 #
1509 #
1510 #
1511 #
1512 #
1513 #
1514 #
1515 #
1516 #
1517 #
1518 #
1519 #
1520 #
1521 #
1522 #
1523 #
1524 #
1525 #
1526 #
1527 #
1528 #
1529 #
1530 #
1531 #
1532 #
1533 #
1534 #
1535 #
1536 #
1537 #
1538 #
1539 #
1540 #
1541 #
1542 #
1543 #
1544 #
1545 #
1546 #
1547 #
1548 #
1549 #
1550 #
1551 #
1552 #
1553 #
1554 #
1555 #
1556 #
1557 #
1558 #
1559 #
1560 #
1561 #
1562 #
1563 #
1564 #
1565 #
1566 #
1567 #
1568 #
1569 #
1570 #
1571 #
1572 #
1573 #
1574 #
1575 #
1576 #
1577 #
1578 #
1579 #
1580 #
1581 #
1582 #
1583 #
1584 #
1585 #
1586 #
1587 #
1588 #
1589 #
1590 #
1591 #
1592 #
1593 #
1594 #
1595 #
1596 #
1597 #
1598 #
1599 #
1600 #
1601 #
1602 #
1603 #
1604 #
1605 #
1606 #
1607 #
1608 #
1609 #
1610 #
1611 #
1612 #
1613 #
1614 #
1615 #
1616 #
1617 #
1618 #
1619 #
1620 #
1621 #
1622 #
1623 #
1624 #
1625 #
1626 #
1627 #
1628 #
1629 #
1630 #
1631 #
1632 #
1633 #
1634 #
1635 #
1636 #
1637 #
1638 #
1639 #
1640 #
1641 #
1642 #
1643 #
1644 #
1645 #
1646 #
1647 #
1648 #
1649 #
1650 #
1651 #
1652 #
1653 #
1654 #
1655 #
1656 #
1657 #
1658 #
1659 #
1660 #
1661 #
1662 #
1663 #
1664 #
1665 #
1666 #
1667 #
1668 #
1669 #
1670 #
1671 #
1672 #
1673 #
1674 #
1675 #
1676 #
1677 #
1678 #
1679 #
1680 #
1681 #
1682 #
1683 #
1684 #
1685 #
1686 #
1687 #
1688 #
1689 #
1690 #
1691 #
1692 #
1693 #
1694 #
1695 #
1696 #
1697 #
1698 #
1699 #
1700 #
1701 #
1702 #
1703 #
1704 #
1705 #
1706 #
1707 #
1708 #
1709 #
1710 #
1711 #
1712 #
1713 #
1714 #
1715 #
1716 #
1717 #
1718 #
1719 #
1720 #
1721 #
1722 #
1723 #
1724 #
1725 #
1726 #
1727 #
1728 #
1729 #
1730 #
1731 #
1732 #
1733 #
1734 #
1735 #
1736 #
1737 #
1738 #
1739 #
1740 #
1741 #
1742 #
1743 #
1744 #
1745 #
1746 #
1747 #
1748 #
1749 #
1750 #
1751 #
1752 #
1753 #
1754 #
1755 #
1756 #
1757 #
1758 #
1759 #
1760 #
1761 #
1762 #
1763 #
1764 #
1765 #
1766 #
1767 #
1768 #
1769 #
1770 #
1771 #
1772 #
1773 #
1774 #
1775 #
1776 #
1777 #
1778 #
1779 #
1780 #
1781 #
1782 #
1783 #
1784 #
1785 #
1786 #
1787 #
1788 #
1789 #
1790 #
1791 #
1792 #
1793 #
1794 #
1795 #
1796 #
1797 #
1798 #
1799 #
1800 #
1801 #
1802 #
1803 #
1804 #
1805 #
1806 #
1807 #
1808 #
1809 #
1810 #
1811 #
1812 #
1813 #
1814 #
1815 #
1816 #
1817 #
1818 #
1819 #
1820 #
1821 #
1822 #
1823 #
1824 #
1825 #
1826 #
1827 #
1828 #
1829 #
1830 #
1831 #
1832 #
1833 #
1834 #
1835 #
1836 #
1837 #
1838 #
1839 #
1840 #
1841 #
1842 #
1843 #
1844 #
1845 #
1846 #
1847 #
1848 #
1849 #
1850 #
1851 #
1852 #
1853 #
1854 #
1855 #
1856 #
1857 #
1858 #
1859 #
1860 #
1861 #
1862 #
1863 #
1864 #
1865 #
1866 #
1867 #
1868 #
1869 #
1870 #
1871 #
1872 #
1873 #
1874 #
1875 #
1876 #
1877 #
1878 #
1879 #
1880 #
1881 #
1882 #
1883 #
1884 #
1885 #
1886 #
1887 #
1888 #
1889 #
1890 #
1891 #
1892 #
1893 #
1894 #
1895 #
1896 #
1897 #
1898 #
1899 #
1900 #
1901 #
1902 #
1903 #
1904 #
1905 #
1906 #
1907 #
1908 #
1909 #
1910 #
1911 #
1912 #
1913 #
1914 #
1915 #
1916 #
1917 #
1918 #
1919 #
1920 #
1921 #
1922 #
1923 #
1924 #
1925 #
1926 #
1927 #
1928 #
1929 #
1930 #
1931 #
1932 #
1933 #
1934 #
1935 #
1936 #
1937 #
1938 #
1939 #
1940 #
1941 #
1942 #
1943 #
1944 #
1945 #
1946 #
1947 #
1948 #
1949 #
1950 #
1951 #
1952 #
1953 #
1954 #
1955 #
1956 #
1957 #
1958 #
1959 #
1960 #
1961 #
1962 #
1963 #
1964 #
1965 #
1966 #
1967 #
1968 #
1969 #
1970 #
1971 #
1972 #
1973 #
1974 #
1975 #
1976 #
1977 #
1978 #
1979 #
1980 #
1981 #
1982 #
1983 #
1984 #
1985 #
1986 #
1987 #
1988 #
1989 #
1990 #
1991 #
1992 #
1993 #
1994 #
1995 #
1996 #
1997 #
1998 #
1999 #
2000 #
2001 #
2002 #
2003 #
2004 #
2005 #
2006 #
2007 #
2008 #
2009 #
2010 #
2011 #
2012 #
2013 #
2014 #
2015 #
2016 #
2017 #
2018 #
2019 #
2020 #
2021 #
2022 #
2023 #
2024 #
2025 #
2026 #
2027 #
2028 #
2029 #
2030 #
2031 #
2032 #
2033 #
2034 #
2035 #
2036 #
2037 #
2038 #
2039 #
2040 #
2041 #
2042 #
2043 #
2044 #
2045 #
2046 #
2047 #
2048 #
2049 #
2050 #
2051 #
2052 #
2053 #
2054 #
2055 #
2056 #
2057 #

```

```

1223 do
1224     case $FLAG in
1225         i ) i_FLAG=y; i_CMD_LINE_FLAG=y
1226            ;;
1227         n ) n_FLAG=y
1228            ;;
1229         S )
1230             set_S_flag $OPTARG
1231             ;;
1232         +t ) t_FLAG=n
1233             ;;
1234         V ) V_FLAG=y
1235             V_ARG="$OPTARG"
1236             ;;
1237         \? ) echo "$USAGE"
1238             exit 1
1239             ;;
1240         esac
1241 done

1243 # correct argument count after options
1244 shift `expr $OPTIND - 1`

1246 # test that the path to the environment-setting file was given
1247 if [ $# -ne 1 ]; then
1248     echo "$USAGE"
1249     exit 1
1250 fi

1252 # check if user is running nightly as root
1253 # ISUSER is set non-zero if an ordinary user runs nightly, or is zero
1254 # when root invokes nightly.
1255 /usr/bin/id | grep '^uid=0(' >/dev/null 2>&1
1256 ISUSER=${?}; export ISUSER

1258 #
1259 # force locale to C
1260 LC_COLLATE=C; export LC_COLLATE
1261 LC_CTYPE=C; export LC_CTYPE
1262 LC_MESSAGES=C; export LC_MESSAGES
1263 LC_MONETARY=C; export LC_MONETARY
1264 LC_NUMERIC=C; export LC_NUMERIC
1265 LC_TIME=C; export LC_TIME

1267 # clear environment variables we know to be bad for the build
1268 unset LD_OPTIONS
1269 unset LD_AUDIT LD_AUDIT_32 LD_AUDIT_64
1270 unset LD_BIND_NOW LD_BIND_NOW_32 LD_BIND_NOW_64
1271 unset LD_BREADTH LD_BREADTH_32 LD_BREADTH_64
1272 unset LD_CONFIG LD_CONFIG_32 LD_CONFIG_64
1273 unset LD_DEBUG LD_DEBUG_32 LD_DEBUG_64
1274 unset LD_DEMANGLE LD_DEMANGLE_32 LD_DEMANGLE_64
1275 unset LD_FLAGS LD_FLAGS_32 LD_FLAGS_64
1276 unset LD_LIBRARY_PATH LD_LIBRARY_PATH_32 LD_LIBRARY_PATH_64
1277 unset LD_LOADFLTR LD_LOADFLTR_32 LD_LOADFLTR_64
1278 unset LD_NOAUDIT LD_NOAUDIT_32 LD_NOAUDIT_64
1279 unset LD_NOAUXFLTR LD_NOAUXFLTR_32 LD_NOAUXFLTR_64
1280 unset LD_NOCONFIG LD_NOCONFIG_32 LD_NOCONFIG_64
1281 unset LD_NODIRCONFIG LD_NODIRCONFIG_32 LD_NODIRCONFIG_64
1282 unset LD_NODIRECT LD_NODIRECT_32 LD_NODIRECT_64
1283 unset LD_NOLAZYLOAD LD_NOLAZYLOAD_32 LD_NOLAZYLOAD_64
1284 unset LD_NOOBJALTER LD_NOOBJALTER_32 LD_NOOBJALTER_64
1285 unset LD_NOVERSION LD_NOVERSION_32 LD_NOVERSION_64
1286 unset LD_ORIGIN LD_ORIGIN_32 LD_ORIGIN_64
1287 unset LD_PRELOAD LD_PRELOAD_32 LD_PRELOAD_64
1288 unset LD_PROFILE LD_PROFILE_32 LD_PROFILE_64

```

```

1290 unset CONFIG
1291 unset GROUP
1292 unset OWNER
1293 unset REMOTE
1294 unset ENV
1295 unset ARCH
1296 unset CLASSPATH
1297 unset NAME

1299 #
1300 # Setup environmental variables
1301 #
1302 if [ -f /etc/nightly.conf ]; then
1303     . /etc/nightly.conf
1304 fi

1306 if [ -f $1 ]; then
1307     if [[ $1 = */* ]]; then
1308         . $1
1309     else
1310         . ./$1
1311     fi
1312 else
1313     if [ -f $OPTHOME/onbld/env/$1 ]; then
1314         . $OPTHOME/onbld/env/$1
1315     else
1316         echo "Cannot find env file as either $1 or $OPTHOME/onbld/env/$1"
1317         exit 1
1318     fi
1319 fi

1321 # contents of stdenv.sh inserted after next line:
1322 # STDENV_START
1323 # STDENV_END

1325 #
1326 # place ourselves in a new task, respecting BUILD_PROJECT if set.
1327 #
1328 if [ -z "$BUILD_PROJECT" ]; then
1329     /usr/bin/newtask -c $$
1330 else
1331     /usr/bin/newtask -c $$ -p $BUILD_PROJECT
1332 fi

1334 ps -o taskid= -p $$ | read build_taskid
1335 ps -o project= -p $$ | read build_project

1337 #
1338 # See if NIGHTLY_OPTIONS is set
1339 #
1340 if [ "$NIGHTLY_OPTIONS" = "" ]; then
1341     NIGHTLY_OPTIONS="-aBm"
1342 fi

1344 #
1345 # If BRINGOVER_WS was not specified, let it default to CLONE_WS
1346 #
1347 if [ "$BRINGOVER_WS" = "" ]; then
1348     BRINGOVER_WS=$CLONE_WS
1349 fi

1351 #
1352 # If CLOSED_BRINGOVER_WS was not specified, let it default to CLOSED_CLONE_WS
1353 #
1354 if [ "$CLOSED_BRINGOVER_WS" = "" ]; then

```

new/usr/src/tools/scripts/nightly.sh

5

```

1355         CLOSED_BRINGOVER_WS=$CLOSED_CLONE_WS
1356 fi

1358 #
1359 # If BRINGOVER_FILES was not specified, default to usr
1360 #
1361 if [ "$BRINGOVER_FILES" = "" ]; then
1362     BRINGOVER_FILES="usr"
1363 fi

1365 #
1366 # If the closed sources are not present, the closed binaries must be
1367 # present for the build to succeed.  If there's no pointer to the
1368 # closed binaries, flag that now, rather than forcing the user to wait
1369 # a couple hours (or more) to find out.
1370 #
1371 orig_closed_is_present="$CLOSED_IS_PRESENT"
1372 check_closed_tree

1374 #
1375 # Note: changes to the option letters here should also be applied to the
1376 # bldenv script.  'd' is listed for backward compatibility.
1377 #
1378 NIGHTLY_OPTIONS=-${NIGHTLY_OPTIONS#-}
1379 OPTIND=1
1380 while getopts AaBCDdFfGgIiLlMmNnOoPpRrS:TtUuWwXxz FLAG $NIGHTLY_OPTIONS
1381 do
1382     case $FLAG in
1383         A ) A_FLAG=y
1384             ;;
1385         a ) a_FLAG=y
1386             ;;
1387         B ) D_FLAG=y
1388             ;; # old version of D
1389         C ) C_FLAG=y
1390             ;;
1391         D ) D_FLAG=y
1392             ;;
1393         F ) F_FLAG=y
1394             ;;
1395         f ) f_FLAG=y
1396             ;;
1397         G ) a_FLAG=y
1398             u_FLAG=y
1399             ;;
1400         I ) a_FLAG=y
1401             m_FLAG=y
1402             p_FLAG=y
1403             u_FLAG=y
1404             ;;
1405         i ) i_FLAG=y
1406             ;;
1407         l ) l_FLAG=y
1408             ;;
1409         M ) M_FLAG=y
1410             ;;
1411         m ) m_FLAG=y
1412             ;;
1413         N ) N_FLAG=y
1414             ;;
1415         n ) n_FLAG=y
1416             ;;
1417         O ) O_FLAG=y
1418             ;;
1419         o ) o_FLAG=y
1420             ;;

```

new/usr/src/tools/scripts/nightly.sh

6

```

1421     P ) P_FLAG=y
1422         ;; # obsolete
1423     p ) p_FLAG=y
1424         ;;
1425     R ) m_FLAG=y
1426         p_FLAG=y
1427         ;;
1428     r ) r_FLAG=y
1429         ;;
1430     S )
1431         set_S_flag $OPTARG
1432         ;;
1433     T ) T_FLAG=y
1434         ;; # obsolete
1435     +t ) t_FLAG=n
1436         ;;
1437     U )
1438         if [ -z "${PARENT_ROOT}" ]; then
1439             echo "PARENT_ROOT must be set if the U flag is" \
1440                 "present in NIGHTLY_OPTIONS."
1441             exit 1
1442         fi
1443         U_FLAG=y
1444         NIGHTLY_PARENT_ROOT=$PARENT_ROOT
1445         ;;
1446     u ) u_FLAG=y
1447         ;;
1448     W ) W_FLAG=y
1449         ;;
1451     w ) w_FLAG=y
1452         ;;
1453     X ) # now that we no longer need realmode builds, just
1454         # copy IHV packages.  only meaningful on x86.
1455         if [ "$MACH" = "i386" ]; then
1456             X_FLAG=y
1457         fi
1458         ;;
1459     x ) XMOD_OPT="-x"
1460         ;;
1461     z ) z_FLAG=y
1462         ;;
1463     \? ) echo "$USAGE"
1464         exit 1
1465         ;;
1466     esac
1467 done

1469 if [ $ISUSER -ne 0 ]; then
1470     if [ "$o_FLAG" = "y" ]; then
1471         echo "Old-style build requires root permission."
1472         exit 1
1473     fi

1475     # Set default value for STAFFER, if needed.
1476     if [ -z "$STAFFER" -o "$STAFFER" = "nobody" ]; then
1477         STAFFER="/usr/xpg4/bin/id -un"
1478         export STAFFER
1479     fi
1480 fi

1482 if [ -z "$MAILTO" -o "$MAILTO" = "nobody" ]; then
1483     MAILTO=$STAFFER
1484     export MAILTO
1485 fi

```

new/usr/src/tools/scripts/nightly.sh

7

```

1487 PATH="$OPTHOME/onbld/bin:$OPTHOME/onbld/bin/${MACH}:/usr/ccs/bin"
1488 PATH="$PATH:$OPTHOME/SUNWSpro/bin:$TEAMWARE/bin:/usr/bin:/usr/sbin:/usr/ucb"
1489 PATH="$PATH:/usr/openwin/bin:/usr/sfw/bin:/opt/sfw/bin:."
1490 export PATH

1492 # roots of source trees, both relative to $SRC and absolute.
1493 relsrcdirs="."
1494 if [[ -d $CODMGR_WS/usr/closed && "$CLOSED_IS_PRESENT" != no ]]; then
1494 if [[ -d $SRC/../closed && "$CLOSED_IS_PRESENT" != no ]]; then
1495     relsrcdirs="$relsrcdirs ../closed"
1496 fi
1497 absrsrcdirs=""
1498 for d in $relsrcdirs; do
1499     absrsrcdirs="$absrsrcdirs $SRC/$d"
1500 done

1502 unset CH
1503 if [ "$o_FLAG" = "y" ]; then
1504 # root invoked old-style build -- make sure it works as it always has
1505 # by exporting 'CH'. The current Makefile.master doesn't use this, but
1506 # the old ones still do.
1507     PROTOCMPTERSE="protocmp.terse"
1508     CH=
1509     export CH
1510 else
1511     PROTOCMPTERSE="protocmp.terse -gu"
1512 fi
1513 POUND_SIGN="#"

1515 # we export POUND_SIGN to speed up the build process -- prevents evaluation of
1516 # the Makefile.master definitions.
1517 export o_FLAG X_FLAG POUND_SIGN

1519 maketype="distributed"
1520 MAKE=dmake
1521 # get the dmake version string alone
1522 DMAKE_VERSION=$( $MAKE -v )
1523 DMAKE_VERSION=${DMAKE_VERSION#*: }
1524 # focus in on just the dotted version number alone
1525 DMAKE_MAJOR=$( echo $DMAKE_VERSION | \
1526     sed -e 's/.*<([^\.]*)\.[^\ ]*\).*$/\1/' )
1527 # extract the second (or final) integer
1528 DMAKE_MINOR=${DMAKE_MAJOR#*.}
1529 DMAKE_MINOR=${DMAKE_MINOR%.*}
1530 # extract the first integer
1531 DMAKE_MAJOR=${DMAKE_MAJOR%.*}
1532 CHECK_DMAKE=${CHECK_DMAKE:-y}
1533 # x86 was built on the 12th, sparc on the 13th.
1534 if [ "$CHECK_DMAKE" = "y" -a \
1535     "$DMAKE_VERSION" != "Sun Distributed Make 7.3 2003/03/12" -a \
1536     "$DMAKE_VERSION" != "Sun Distributed Make 7.3 2003/03/13" -a \ \
1537     "$DMAKE_MAJOR" -lt 7 -o \
1538     "$DMAKE_MAJOR" -eq 7 -a "$DMAKE_MINOR" -lt 4 \ ] ]; then
1539     if [ -z "$DMAKE_VERSION" ]; then
1540         echo "$MAKE is missing."
1541         exit 1
1542     fi
1543     echo `whence $MAKE` version is:
1544     echo " ${DMAKE_VERSION}"
1545     cat <<EOF

1547 This version may not be safe for use. Either set TEAMWARE to a better
1548 path or (if you really want to use this version of dmake anyway), add
1549 the following to your environment to disable this check:

1551 CHECK_DMAKE=n

```

new/usr/src/tools/scripts/nightly.sh

8

```

1552 EOF
1553     exit 1
1554 fi
1555 export PATH
1556 export MAKE

1558 if [ "${SUNWSPRO}" != "" ]; then
1559     PATH="${SUNWSPRO}/bin:$PATH"
1560     export PATH
1561 fi

1563 hostname=`uname -n`
1564 if [ ! -f $HOME/.make.machines ]; then
1565     DMAKE_MAX_JOBS=4
1566 else
1567     DMAKE_MAX_JOBS=`grep $hostname $HOME/.make.machines | \
1568         tail -1 | awk -F= '{print $2;}'`
1569     if [ "$DMAKE_MAX_JOBS" = "" ]; then
1570         DMAKE_MAX_JOBS=4
1571     fi
1572 fi
1573 DMAKE_MODE=parallel;
1574 export DMAKE_MODE
1575 export DMAKE_MAX_JOBS

1577 if [ -z "${ROOT}" ]; then
1578     echo "ROOT must be set."
1579     exit 1
1580 fi

1582 #
1583 # if -V flag was given, reset VERSION to V_ARG
1584 #
1585 if [ "$V_FLAG" = "y" ]; then
1586     VERSION=$V_ARG
1587 fi

1589 #
1590 # Check for IHV root for copying ihv proto area
1591 #
1592 if [ "$X_FLAG" = "y" ]; then
1593     if [ "$IA32_IHV_ROOT" = "" ]; then
1594         echo "IA32_IHV_ROOT: must be set for copying ihv proto"
1595         args_ok=n
1596     fi
1597     if [ ! -d "$IA32_IHV_ROOT" ]; then
1598         echo "$IA32_IHV_ROOT: not found"
1599         args_ok=n
1600     fi
1601     if [ "$IA32_IHV_WS" = "" ]; then
1602         echo "IA32_IHV_WS: must be set for copying ihv proto"
1603         args_ok=n
1604     fi
1605     if [ ! -d "$IA32_IHV_WS" ]; then
1606         echo "$IA32_IHV_WS: not found"
1607         args_ok=n
1608     fi
1609 fi

1611 # Append source version
1612 if [ "$SSE_FLAG" = "y" ]; then
1613     VERSION="${VERSION}:EXPORT"
1614 fi

1616 if [ "$SSD_FLAG" = "y" ]; then
1617     VERSION="${VERSION}:DOMESTIC"

```

```

1618 fi
1620 if [ "$SSH_FLAG" = "y" ]; then
1621     VERSION="${VERSION}:MODIFIED_SOURCE_PRODUCT"
1622 fi
1624 if [ "$SO_FLAG" = "y" ]; then
1625     VERSION="${VERSION}:OPEN_ONLY"
1626 fi
1628 TMPDIR="/tmp/nightly.tmpdir.$$"
1629 export TMPDIR
1630 rm -rf ${TMPDIR}
1631 mkdir -p $TMPDIR || exit 1
1633 #
1634 # Keep elfsign's use of pkcs11_softtoken from looking in the user home
1635 # directory, which doesn't always work. Needed until all build machines
1636 # have the fix for 6271754
1637 #
1638 SOFTTOKEN_DIR=$TMPDIR
1639 export SOFTTOKEN_DIR
1641 TOOLS=${SRC}/tools
1642 TOOLS_PROTO=${TOOLS}/proto
1644 unset CFLAGS LD_LIBRARY_PATH LDFLAGS
1646 # create directories that are automatically removed if the nightly script
1647 # fails to start correctly
1648 newdir() {
1649     dir=$1
1650     toadd=
1651     while [ ! -d $dir ]; do
1652         toadd="$dir $toadd"
1653         dir=`dirname $dir`
1654     done
1655     torm=
1656     newlist=
1657     for dir in $toadd; do
1658         if staffer mkdir $dir; then
1659             newlist="$ISUSER $dir $newlist"
1660             torm="$dir $torm"
1661         else
1662             [ -z "$torm" ] || staffer rmdir $torm
1663             return 1
1664         fi
1665     done
1666     newdirlist="$newlist $newdirlist"
1667     return 0
1668 }
1669 unchanged_portion_omitted
2138 type bringover_mercurial > /dev/null 2>&1 || bringover_mercurial() {
2139     typeset -x PATH=$PATH
2141     # If the repository doesn't exist yet, then we want to populate it.
2142     # We only initially populate the closed repository if the open
2143     # repository doesn't also exist (e.g.: don't bringover closed if
2144     # only usr/src exists)
2142     if [[ ! -d $CODEMGR_WS/.hg ]]; then
2143         staffer hg init $CODEMGR_WS
2144     fi
2146     #
2147     # If the user set CLOSED_BRINGOVER_WS and didn't set CLOSED_IS_PRESENT

```

```

2148     # to "no," then we'll want to initialise the closed repository
2149     #
2150     # We use $orig_closed_is_present instead of $CLOSED_IS_PRESENT,
2151     # because for newly-created source trees, the latter will be "no"
2152     # until after the bringover completes.
2153     #
2154     if [[ "$orig_closed_is_present" != "no" && \
2155         -n "$CLOSED_BRINGOVER_WS" && \
2156         ! -d $CODEMGR_WS/usr/closed/.hg ]]; then
2157         staffer mkdir -p $CODEMGR_WS/usr/closed
2158         # If the user set CLOSED_BRINGOVER_WS, then we'll want to
2159         # initialise the closed repository
2160         if [[ -n $CLOSED_BRINGOVER_WS ]]; then
2161             staffer mkdir -p $CODEMGR_WS/usr
2162             staffer hg init $CODEMGR_WS/usr/closed
2163             export CLOSED_IS_PRESENT=yes
2164             CLOSED_IS_PRESENT="yes"
2165         fi
2166     fi
2162     typeset -x HGMERGE="/bin/false"
2164     #
2165     # Do the bringover, along with an update. If a merge is necessary,
2166     # this will not fail properly:
2167     # http://www.selenic.com/mercurial/bts/issue186
2168     # so we grep through the output for signs of a merge.
2169     #
2170     # Note that pulling even non-conflicting changes on top of local
2171     # changes (committed or not) requires a merge.
2172     #
2173     staffer hg --cwd $CODEMGR_WS pull -u $BRINGOVER_WS 2>&1 | \
2174         tee $TMPDIR/pull_open.out
2175     hgstatus=$?
2176     if [ $hgstatus -ne 0 ]; then
2177         touch $TMPDIR/bringover_failed
2178         return
2179     fi
2181     #
2182     # Again, we only want to update usr/closed if it exists, and we
2183     # haven't been told not to via $CLOSED_IS_PRESENT, and we actually
2184     # know where to pull from ($CLOSED_BRINGOVER_WS).
2185     #
2186     if [[ $CLOSED_IS_PRESENT = yes && \
2187         -d $CODEMGR_WS/usr/closed/.hg && \
2188         -n $CLOSED_BRINGOVER_WS ]]; then
2189         staffer hg --cwd $CODEMGR_WS/usr/closed pull -u \
2190             # We never want to clone usr/closed by default. If the usr/closed
2191             # repository exists already and CLOSED_BRINGOVER_WS is set, then
2192             # we update (pull -u).
2193             if [[ -d $SRC/../closed && -n $CLOSED_BRINGOVER_WS ]]; then
2194                 staffer hg --cwd $SRC/../closed pull -u \
2195                     $CLOSED_BRINGOVER_WS 2>&1 |
2196                     tee -a $TMPDIR/pull_closed.out
2197             hgstatus=$?
2198             if [ $hgstatus -ne 0 ]; then
2199                 touch $TMPDIR/bringover_failed
2200                 return
2201             fi
2202     fi
2199     #
2200     # If there were uncommitted changes, the pull -u would have
2201     # attempted a merge. In the likely event that it failed, fail the
2202     # bringover.

```

```

2203 #
2204 if grep "^merging.*failed" $TMPDIR/pull_open.out > /dev/null 2>&1 || \
2205     grep "^merging.*failed" $TMPDIR/pull_closed.out >/dev/null 2>&1;
2206 then
2207     touch $TMPDIR/bringover_failed
2208     return
2209 fi

2211 # If heads were added, then a merge is required. We attempt the
2212 # merge, in case the branches touch different files (something that
2213 # teamware wouldn't balk at). If the merge fails, fail the
2214 # bringover, leaving the conflicts unresolved.
2215 #
2216 # Note that even if the merge succeeds, the result of the merge
2217 # will need to be committed, and not doing so will block any
2218 # further pulls.

2220 typeset mergefailfmt="\
2221 ***\n\
2222 *** In repository %s:\n\
2223 ***\n\
2224 *** nightly was unable to automatically merge your changes.\n\
2225 *** Run hg merge manually, then restart nightly with -n.\n\
2226 ***\n"

2228 typeset mergepassfmt="\
2229 ***\n\
2230 *** In repository %s:\n\
2231 ***\n\
2232 *** nightly successfully merged your changes. This means that your working\n\
2233 *** directory has been updated, but those changes are not yet committed.\n\
2234 *** After nightly completes, you should validate the results of the merge,\n\
2235 *** then use hg commit manually.\n\
2236 ***\n"

2238 if grep "not updating" $TMPDIR/pull_open.out > /dev/null 2>&1; then
2239     hg --cwd $CODEMGR_WS merge 2>&1 | tee $TMPDIR/merge.out
2240     if grep "^merging.*failed" $TMPDIR/merge.out >/dev/null 2>&1;
2241     then
2242         printf "$mergefailfmt" "${CODEMGR_WS}"
2243         touch $TMPDIR/bringover_failed
2244     else
2245         printf "$mergepassfmt" "${CODEMGR_WS}"
2246     fi
2247 fi
2248 if grep "not updating" $TMPDIR/pull_closed.out > /dev/null 2>&1; then
2249     hg --cwd $CODEMGR_WS/usr/closed merge 2>&1 | tee $TMPDIR/merge.o
2213     hg --cwd $SRC/./closed merge 2>&1 | tee $TMPDIR/merge.out
2250     if grep "^merging.*failed" $TMPDIR/merge.out >/dev/null 2>&1;
2251     then
2252         printf "$mergefailfmt" "${CODEMGR_WS}/usr/closed"
2253         touch $TMPDIR/bringover_failed
2254     else
2255         printf "$mergepassfmt" "${CODEMGR_WS}/usr/closed"
2256     fi
2257 fi
2258 }

```

unchanged portion omitted