```
Last login: Sat Apr 15 13:50:34 on ttys001
[wansongHome@wansong:~]$ vim test.sh
[wansongHome@wansong:~]$ vim test.sh
[wansongHome@wansong:~]$ add
-bash: add: command not found
[wansongHome@wansong:~]$ source  test.sh
we are being sourced
[wansongHome@wansong:~]$ add 1 2
(( 1 + 2 ))
[wansongHome@wansong:~]$ vim test.sh
[wansongHome@wansong:~]$ source test.sh
-bash: test.sh: line 4: syntax error near unexpected token `('
-bash: test.sh: line 4: `  echo (( $1 + $2 ))'
[wansongHome@wansong:~]$ vim test.sh
[wansongHome@wansong:~]$ source test.sh
-bash: test.sh: line 4: syntax error near unexpected token `('
-bash: test.sh: line 4: `  ret=(($1+$2))'

-bash: test.sh: line 6: syntax error near unexpected token `}'
-bash: test.sh: line 6: `}'
[wansongHome@wansong:~]$ vim test.sh
[wansongHome@wansong:~]$ mab  bash
-bash: mab: command not found
[wansongHome@wansong:~]$ man bash
[wansongHome@wansong:~]$ vim test.sh
[wansongHome@wansong:~]$ source test.sh
we are being sourced
[wansongHome@wansong:~]$ add 1 2
3
[wansongHome@wansong:~]$ vim test.sh
[wansongHome@wansong:~]$ source test.sh
we are being sourced
[wansongHome@wansong:~]$ add 1 2
3
[wansongHome@wansong:~]$ vim test.sh
[wansongHome@wansong:~]$ man bash
[wansongHome@wansong:~]$ man bash
[wansongHome@wansong:~]$ cd /usr/
[wansongHome@wansong:/usr]$ ls
X11  X11R6  adic  bin  include  lib  libexec  local  sbin  share  standalone
[wansongHome@wansong:/usr]$ ls lib
bundle1.o                       libbz2.1.0.5.dylib       libform.5.4.dylib         libnetsnmp.15.dylib         libsasl2.2.0.21.dylib
charset.alias                   libbz2.1.0.dylib         libform.dylib             libnetsnmp.25.dylib         libsasl2.2.0.22.dylib
cron                            libbz2.dylib             libgcc_s.1.dylib          libnetsnmp.5.2.1.dylib      libsasl2.2.dylib
crt1.10.5.o                     libc++.1.dylib           libgcc_s.10.4.dylib       libnetsnmp.5.dylib          libsasl2.dylib
crt1.10.6.o                     libc++.dylib             libgcc_s.10.5.dylib       libnetsnmp.dylib            libspindump.dylib
crt1.o                          libc++abi.dylib          libgermantok.dylib        libnetsnmpagent.25.dylib    libsqlite3.0.dylib
dtrace                          libc.dylib               libgmalloc.dylib          libnetsnmpagent.dylib       libsqlite3.dylib
dyld                            libcblas.dylib           libgssapi_krb5.dylib      libnetsnmphelpers.25.dylib  libssl.0.9.7.dylib
dylib1.10.5.o                   libcharset.1.0.0.dylib   libgutenprint.2.dylib     libnetsnmphelpers.dylib     libssl.0.9.8.dylib
dylib1.o                        libcharset.1.dylib       libheimdal-asn1.dylib     libnetsnmpmibs.25.dylib     libssl.35.dylib
gcrt1.o                         libcharset.dylib         libhunspell-1.2.0.0.0.dylib libnetsnmpmibs.dylib      libssl.dylib
groff                           libclapack.dylib         libhunspell-1.2.0.dylib   libnetsnmptrapd.25.dylib    libstdc++.6.0.9.dylib
lazydylib1.o                    libcmph.dylib            libhunspell-1.2.dylib     libnetsnmptrapd.dylib       libstdc++.6.dylib
libATCommandStudio.a            libcom_err.dylib         libiconv.2.4.0.dylib      libnetwork.dylib            libstdc++.dylib
libATCommandStudioDynamic.dylib libcompression.dylib     libiconv.2.dylib          libobjc.A.dylib            libsysmon.dylib
libAVFAudio.dylib               libcrypto.0.9.7.dylib    libiconv.dylib            libobjc.dylib              libsystemstats.dylib
libAccountPolicyTranslation.dylib libcrypto.0.9.8.dylib  libicucore.A.dylib        libodfde.dylib             libtcl.dylib
libBSDPClient.A.dylib           libcrypto.35.dylib       libicucore.dylib          libodmodule.dylib          libtcl8.5.dylib
libBSDPClient.dylib             libcrypto.dylib          libinfo.dylib             libpam.1.dylib             libtclstub8.5.a
libCRFSuite.dylib               libcsfde.dylib           libiodbc.2.1.18.dylib     libpam.2.dylib             libtermcap.dylib
libCRFSuite0.12.dylib           libcups.2.dylib          libiodbc.2.dylib          libpam.dylib               libtidy.A.dylib
libChineseTokenizer.dylib       libcups.dylib            libiodbc.dylib            libpanel.5.4.dylib         libtidy.dylib
libCoreStorage.dylib            libcupscgi.1.dylib       libiodbcinst.2.1.18.dylib  libpanel.dylib            libtk.dylib
libDHCPServer.A.dylib           libcupscgi.dylib         libiodbcinst.2.dylib      libpcap.A.dylib            libtk8.5.dylib
libDHCPServer.dylib             libcupsimage.2.dylib     libiodbcinst.dylib        libpcap.dylib              libtkstub8.5.a
libDiagnosticMessagesClient.dylib libcupsimage.dylib     libipconfig.dylib         libpcre.0.dylib           libtls.6.dylib
libFosl_dynamic.dylib           libcupsmime.1.dylib      libipsec.A.dylib          libpcre.dylib             libtls.dylib
libIASAuthReboot.dylib          libcupsmime.dylib        libipsec.dylib            libpcreposix.0.dylib       libutil.dylib
libIASUnifiedProgress.dylib     libcupsppdc.1.dylib      libk5crypto.dylib         libpcreposix.dylib         libutil1.0.dylib
libMatch.1.dylib                libcupsppdc.dylib        libkmod.a                 libpgtypes.3.4.dylib       libxar.1.dylib
libMatch.dylib                  libcurl.3.dylib          libkmodc++.a              libpgtypes.3.dylib         libxar.dylib
libOpenScriptingUtil.dylib      libcurl.4.dylib          libkrb4.dylib             libpgtypes.dylib           libxcselect.dylib
libQMIParser.a                  libcurl.dylib            libkrb5.dylib             libpkstart.a               libxml2.2.dylib
libQMIParserDynamic.dylib       libcurses.dylib          libkrb524.dylib           libpmenergy.dylib          libxml2.dylib
libScreenReader.dylib           libdbm.dylib             libkrb5support.dylib      libpmsample.dylib          libxslt.1.dylib
libSystem.B.dylib               libdes425.dylib          libktrace.dylib           libpoll.dylib             libxslt.dylib
libSystem.dylib                 libdl.dylib              libl.a                    libpq.5.6.dylib            liby.a
libTelephonyUtilDynamic.dylib   libdns_services.dylib    liblangid.dylib           libpq.5.dylib              libz.1.1.3.dylib
libThaiTokenizer.dylib          libdtrace.dylib          liblapack.dylib           libpq.dylib               libz.1.2.5.dylib
libUniversalAccess.dylib        libecpg.6.5.dylib        liblber.dylib             libprequelite.dylib        libz.1.dylib
libXSEvent.dylib                libecpg.6.dylib          libldap.dylib             libproc.dylib             libz.dylib
libXplugin.1.dylib              libecpg.dylib            libldap_r.dylib           libpthread.dylib           pam
libXplugin.dylib                libecpg_compat.3.5.dylib liblzma.5.dylib           libpython.dylib            php
libalias.A.dylib                libecpg_compat.3.dylib   liblzma.dylib             libpython2.6.dylib         pkgconfig
libalias.dylib                  libecpg_compat.dylib     libm.dylib                libpython2.7.dylib         python2.6
libapr-1.0.dylib                libedit.2.dylib          libmarisa.dylib           libquit.dylib              python2.7
libapr-1.dylib                  libedit.3.0.dylib        libmecab.1.0.0.dylib      libreadline.dylib          rpcsvc
libaprutil-1.0.dylib            libedit.3.dylib          libmecab.dylib            libresolv.9.dylib          ruby
libaprutil-1.dylib              libedit.dylib            libmecabra.dylib          libresolv.dylib            sa
libarchive.2.dylib              libenergytrace.dylib     libmenu.5.4.dylib         librpcsvc.dylib            sasl2
libarchive.dylib                libexpat.1.dylib         libmenu.dylib             libruby.2.0.0.dylib        sqlite3
libauditd.0.dylib               libexpat.dylib           libmx.A.dylib             libruby.2.0.dylib          ssh-keychain.dylib
libauditd.dylib                 libexslt.0.dylib         libmx.dylib               libruby.dylib             system
libauto.dylib                   libexslt.dylib           libncurses.5.4.dylib      libsandbox.1.dylib         tclConfig.sh
libblas.dylib                   libextension.dylib       libncurses.5.dylib        libsandbox.dylib           tkConfig.sh
libbsm.0.dylib                  libf77lapack.dylib       libncurses.dylib          libsasl2.2.0.1.dylib       xpc
libbsm.dylib                    libffi.dylib             libnetsnmp.15.1.2.dylib   libsasl2.2.0.15.dylib      zsh
[wansongHome@wansong:/usr]$ ls
X11  X11R6  adic  bin  include  lib  libexec  local  sbin  share  standalone
[wansongHome@wansong:/usr]$ ls share/
CSI                         cracklib       doc        ftpd        httpd  kdrl.bundle  man       pmenergy  skel      texinfo            uucp
CoreDuetDaemonConfig.bundle cups           emacs      germantok   icu    kpep         mecabra   ri        snmp      thermald.bundle    vim
apr-1                       current-os.sdk examples   groff       info   langid       misc      sandbox   tabset    tokenizer          zoneinfo
calendar                    dict           file       hiutil      java   locale       php       screen    terminfo  ucupdate           zsh
[wansongHome@wansong:/usr]$ cd share/man/
[wansongHome@wansong:/usr/share/man]$ ls
man1  man2  man3  man4  man5  man6  man7  man8  man9  mann  whatis
[wansongHome@wansong:/usr/share/man]$ ls man5/
BridgeSupport.5      crontab.5           hosts.5           networks.5       shells.5          slapo-constraint.5  ssh_config.5
access.5             cups-files.conf.5.gz hosts.equiv.5    newsyslog.conf.5 slapd-bdb.5       slapo-dds.5         sshd_config.5
acct.5               cups-snmp.conf.5.gz info.5            nfs.conf.5       slapd-config.5    slapo-dyngroup.5    subscriptions.conf.5.gz
aliases.5            cupsd-logs.5.gz     intro.5           nisplus_table.5  slapd-dnssrv.5    slapo-dynlist.5     sudoers.5
asl.conf.5           cupsd.conf.5.gz     ipptoolfile.5.gz  nologin.5        slapd-hdb.5       slapo-memberof.5    sysctl.conf.5
audit.5              dir.5               krb5.conf.5       nsmb.conf.5      slapd-ldap.5      slapo-pbind.5       syslog.conf.5
audit.log.5          dirent.5            lastlog.5         pam.conf.5       slapd-ldbm.5      slapo-pcache.5      tcp_table.5
audit_class.5        editrc.5            launchd.plist.5   passwd.5         slapd-ldif.5      slapo-ppolicy.5     term.5
audit_control.5      emond.plist.5       ldap.conf.5       pcap-savefile.5  slapd-mdb.5       slapo-refint.5      terminfo.5
audit_event.5        euc.5               ldap_table.5      pcre_table.5     slapd-meta.5      slapo-retcode.5     texinfo.5
audit_user.5         exports.5           ldif.5            periodic.conf.5  slapd-monitor.5   slapo-rwm.5         tkconrc.5
audit_warn.5         finger.conf.5       lj4_font.5        pf.conf.5        slapd-ndb.5       slapo-sock.5        transport.5
auto_master.5        fstab.5             lmdb_table.5      pf.os.5          slapd-null.5      slapo-sssvlv.5      ttys.5
```

```
autofs.conf.5             ftpchroot.5             magic.5                 pgsql_table.5       slapd-passwd.5      slapo-syncprov.5     types.5
big5.5                    ftpd.conf.5             mailto.conf.5.gz        plist.5             slapd-perl.5        slapo-translucent.5  tzfile.5
body_checks.5            ftpusers.5              man.conf.5              postconf.5          slapd-relay.5       slapo-unique.5       unbound.conf.5
bom.5                     gb18030.5               master.5                postfix-wrapper.5   slapd-shell.5       slapo-valsort.5      utf2.5
bootparams.5             gb2312.5                master.passwd.5         ppdcfile.5.gz       slapd-sock.5        snfs.cfg.5           utf8.5
bootptab.5               gbk.5                   memcache_table.5        printers.conf.5.gz  slapd-sql.5         snfs.cfgx.5          utmp.5
bounce.5                  generic.5               mib2c.conf.5            protocols.5         slapd.access.5      snfs_config.5        utmpx.5
canonical.5              gettytab.5              mime.convs.5.gz         racoon.conf.5       slapd.backends.5    snmp.conf.5          uuencode.5
cidr_table.5            groff_font.5            mime.types.5.gz         regexp_table.5      slapd.conf.5        snmp_config.5        variables.5
classes.conf.5.gz        groff_out.5             moduli.5                relocated.5         slapd.overlays.5    snmpd.conf.5         virtual.5
client.conf.5.gz         groff_tmac.5            motd.5                  resolver.5          slapd.plugin.5      snmpd.examples.5     vpnd.5
com.apple.Boot.plist.5   group.5                 mskanji.5               rpc.5               slapo-accesslog.5   snmpd.internal.5     wtmp.5
compat.5                  gss.5                   mysql_table.5           rsyncd.conf.5.gz    slapo-auditlog.5    snmptrapd.conf.5     x509v3_config.5ssl
config.5ssl              header_checks.5         nanorc.5                rtadvd.conf.5       slapo-chain.5       socketmap_table.5    xpcservice.plist.5
core.5                    heimdal_debug.5         netgroup.5              services.5          slapo-collect.5     sqlite_table.5       xprintf.5
[wansongHome@wansong:/usr/share/man]$ find ./ -iname '*bash*'
./man1/bash.1
./man1/bashbug.1
[wansongHome@wansong:/usr/share/man]$ cd man1
[wansongHome@wansong:/usr/share/man/man1]$ ls
AppleFileServer.1        eyapp5.18.1             mklocale.1                      perldsc5.16.1       perlsub5.18.1        serialver.1
CA.pl.1ssl               false.1                 mktemp.1                        perldsc5.18.1       perlsymbian.1        servertool.1
GoogleSyncMigrate.1      fax.1                   mmroff.1                        perldtrace.1        perlsymbian5.16.1    sess_id.1ssl
KeychainMigrator.1       fc.1                    mnthome.1                       perldtrace5.16.1    perlsymbian5.18.1    set.1
WirelessRadioManagerd.1  fddist.1m               more.1.gz                       perldtrace5.18.1    perlsyn.1            setenv.1
'[.1'                    fg.1                    mpioutil.1                      perlebcdic.1        perlsyn5.16.1        setquota.1
a2p.1                    fgrep.1                 mptcp_client.1                  perlebcdic5.16.1    perlsyn5.18.1        settc.1
a2p.1.gz                 fi.1                    msgs.1                          perlebcdic5.18.1    perlthanks.1         setty.1
a2p5.16.1                file.1                  mv.1                            perlembed.1         perlthanks5.16.1     setuids.d.1m
a2p5.18.1                filebyproc.d.1m         nano.1                          perlembed5.16.1     perlthanks5.18.1     setvar.1
ab.1                     filetest.1              native2ascii.1                  perlembed5.18.1     perlthrtut.1         sftp.1
accesstool.1             filtercalltree.1        nc.1                            perlepoc.1          perlthrtut5.16.1     sh.1
addftinfo.1              find.1                  ncal.1                          perlepoc5.16.1      perlthrtut5.18.1     shar.1
afconvert.1              find2perl.1             ncctl.1                         perlexperiment.1    perltie.1            sharing.1
afhash.1                 find2perl5.16.1         ncdestroy.1                     perlexperiment5.16.1 perltie5.16.1       shasum.1
afida.1                  find2perl5.18.1         ncinit.1                        perlexperiment5.18.1 perltie5.18.1       shasum5.16.1
afinfo.1                 findrule.1              nclist.1                        perlfaq.1           perltoc.1            shasum5.18.1
afmtodit.1               findrule5.16.1          ncurses5.4-config.1             perlfaq1.1          perltoc5.16.1        shift.1
afplay.1                 findrule5.18.1          neqn.1                          perlfaq15.16.1      perltoc5.18.1        shlock.1
afscexpand.1             finger.1                net-server.1                    perlfaq15.18.1      perltodo.1           sigdist.d.1m
agentxtrap.1             fixproc.1               net-server5.16.1                perlfaq2.1          perltodo5.16.1       sips.1
alias.1                  fmt.1                   net-server5.18.1                perlfaq25.16.1      perltodo5.18.1       sleep.1
alloc.1                  fold.1                  net-snmp-config.1               perlfaq25.18.1      perltooc.1           slogin.1
applesingle.1            footprint.1             net-snmp-create-v3-user.1       perlfaq3.1          perltooc5.16.1       smbutil.1
appletviewer.1           for.1                   netlog.1                        perlfaq35.16.1      perltooc5.18.1       smcDiagnose.1
apply.1                  foreach.1               netstat.1                       perlfaq35.18.1      perltoot.1           smime.1ssl
apropos.1                from.1                  nettop.1                        perlfaq4.1          perltoot5.16.1       sndiskmove.1
apt.1                    fs_usage.1              newaliases.1                    perlfaq45.16.1      perltoot5.18.1       snfsdefrag.1
apxs.1                   ftp.1                   newgrp.1                        perlfaq45.18.1      perltrap.1           snmp-bridge-mib.1
arch.1                   funzip.1                newproc.d.1m                    perlfaq5.1          perltrap5.16.1       snmpbulkget.1
asctl.1                  fuser.1                 nfsstat.1                       perlfaq5.16.1       perltrap5.18.1       snmpbulkwalk.1
asn1parse.1ssl           fwkdp.1                 nice.1                          perlfaq5.18.1       perltru64.1          snmpcmd.1
assetutil.1              fwkpfv.1                nl.1                            perlfaq55.16.1      perltru645.16.1      snmpconf.1
at.1                     gdiffmk.1               nohup.1                         perlfaq55.18.1      perltru645.18.1      snmpdelta.1
atos.1                   gem.1                   notify.1                        perlfaq6.1          perltw.1             snmpdf.1
atq.1                    gen_bridge_metadata.1   notifyutil.1                    perlfaq65.16.1      perltw5.16.1         snmpget.1
atrm.1                   gencat.1                nroff.1                         perlfaq65.18.1      perltw5.18.1         snmpgetnext.1
auditreduce.1            gendsa.1ssl             nseq.1ssl                       perlfaq7.1          perlunicode.1        snmpinform.1
authopen.1               genrsa.1ssl             nslookup.1                      perlfaq75.16.1      perlunicode5.16.1    snmpnetstat.1
automator.1              getconf.1               nsupdate.1                      perlfaq75.18.1      perlunicode5.18.1    snmpset.1
auval.1                  getopt.1                ntp-keygen.1                    perlfaq8.1          perlunifaq.1         snmpstatus.1
auvaltool.1              getopts.1               ntpd.1                          perlfaq85.16.1      perlunifaq5.16.1     snmptable.1
avbdeviced.1             glob.1                  ntpdc.1                         perlfaq85.18.1      perlunifaq5.18.1     snmptest.1
avbdiagnose.1            gnuattach.1             ntpq.1                          perlfaq9.1          perluniintro.1       snmptranslate.1
avbutil.1                gnuclient.1             ocsp.1ssl                       perlfaq95.16.1      perluniintro5.16.1   snmptrap.1
avconvert.1              gnudoit.1               ocspd.1                         perlfaq95.18.1      perluniintro5.18.1   snmpusm.1
awk.1                    gnuserv.1               od.1                            perlfilter.1        perluniprops.1       snmpvacm.1
axreg.1                  goto.1                  odmigrationtool.1               perlfilter5.16.1    perluniprops5.16.1   snmpwalk.1
base64.1                 grap2graph.1            odutil.1                        perlfilter5.18.1    perluniprops5.18.1   sntp.1
basename.1               graphicssession.1       onintr.1                        perlfork.1          perlunitut.1         soelim.1
bash.1                   grep.1                  open.1                          perlfork5.16.1      perlunitut5.16.1     sort.1
bashbug.1                grn.1                   opensnoop.1m                    perlfork5.18.1      perlunitut5.18.1     source.1
batch.1                  grodvi.1                openssl.1ssl                    perlform.1          perlutil.1           speed.1ssl
bc.1                     groff.1                 orbd.1                          perlform5.16.1      perluti15.16.1       spfd.1
bg.1                     groffer.1               osacompile.1                    perlform5.18.1      perluti15.18.1       spfd5.16.1
biff.1                   grog.1                  osadecompile.1                  perlfreebsd.1       perluts.1            spfd5.18.1
bind.1                   grohtml.1               osalang.1                       perlfreebsd5.16.1   perluts5.16.1        spfquery.1
bindkey.1                grolbp.1                osascript.1                     perlfreebsd5.18.1   perlvar.1            spfquery5.16.1
binhex.1                 grolj4.1                ostraceutil.1                   perlfunc.1          perlvar5.16.1        spfquery5.18.1
binhex.pl.1              grops.1                 pack200.1                       perlfunc5.16.1      perlvar5.18.1        spkac.1ssl
binhex.pl5.18.1          grotty.1                package-stash-conflicts.1       perlfunc5.18.1      perlvmesa.1          splain.1
bitesize.d.1m            groups.1                package-stash-conflicts5.18.1   perlgit.1           perlvmesa5.16.1      splain5.16.1
brctl.1                  gunzip.1                pagesize.1                      perlgit5.16.1       perlvms.1            splain5.18.1
break.1                  gzcat.1                 par.pl.1                        perlgit5.18.1       perlvms5.16.1        split.1
breaksw.1                gzexe.1                 par.pl5.16.1                    perlglossary.1      perlvms5.18.1        srm.1
bsdtar.1                 gzip.1                  par.pl5.18.1                    perlglossary5.16.1  perlvos.1            ssh-add.1
bspatch.1                h2ph.1                  parl.1                          perlglossary5.18.1  perlvos5.16.1        ssh-agent.1
builtin.1                h2ph5.16.1              parl5.16.1                      perlgpl.1           perlvos5.18.1        ssh-keygen.1
builtins.1               h2ph5.18.1              parl5.18.1                      perlgpl5.16.1       perlwin32.1          ssh-keyscan.1
bunzip2.1                h2xs.1                  passwd.1                        perlgpl5.18.1       perlwin325.16.1      ssh.1
bzcat.1                  h2xs5.16.1              passwd.1ssl                     perlguts.1          perlwin325.18.1      stat.1
bzcmp.1                  h2xs5.18.1              paste.1                         perlguts5.16.1      perlxs.1             stop.1
bzdiff.1                 hash.1                  patch.1                         perlguts5.18.1      perlxs5.16.1         stringdups.1
bzegrep.1                hashstat.1              pathchk.1                       perlhack.1          perlxs5.18.1         stringdups32.1
bzfgrep.1                hdiutil.1               pathopens.d.1m                  perlhack5.16.1      perlxstut.1          stty.1
bzgrep.1                 head.1                  pax.1                           perlhack5.18.1      perlxstut5.16.1      su.1
bzip2.1                  heap.1                  pbcopy.1                        perlhacktips.1      perlxstut5.18.1      sum.1
bzip2recover.1           heap32.1                pbpaste.1                       perlhacktips5.16.1  perlxstypemap.1      suspend.1
bzless.1                 hexdump.1               pcap-config.1                   perlhacktips5.18.1  perlxstypemap5.16.1  sw_vers.1
bzmore.1                 history.1               pcsstatus.1                     perlhacktut.1       perlxstypemap5.18.1  switch.1
c++filt.1                hiutil.1                perl.1                          perlhacktut5.16.1   pfbtops.1            symbols.1.gz
c2ph.1                   host.1                  perl.1.gz                       perlhacktut5.18.1   pgrep.1              symbolscache.1.gz
c2ph5.16.1               hostname.1              perl5.16.1                      perlhaiku.1         phar.1               syscallbypid.d.1m
c2ph5.18.1               hotspot.d.1m            perl5.16.3.1                    perlhaiku5.16.1     phar.phar.1          syscallbyproc.d.1m
c_rehash.1ssl            hpftodit.1              perl5.16.35.16.1                perlhaiku5.18.1     php-config.1         syscallbysysc.d.1m
ca.1ssl                  htdbm.1                 perl5.165.16.1                  perlhist.1          php.1                sysdiagnose.1
cal.1                    htdigest.1              perl5.18.1                      perlhist5.16.1      phpize.1             syslog.1
calendar.1               htpasswd.1              perl5.18.2.1                    perlhist5.18.1      pic.1                systemsoundserverd.1
cancel.1.gz              httpdstat.d.1m          perl5.18.25.18.1                perlhpux.1          pic2graph.1          tab2space.1
cap_mkdb.1               httxt2dbm.1             perl5.185.18.1                  perlhpux5.16.1      pico.1               tabs.1
captoinfo.1m             hup.1                   perl5004delta.1                 perlhpux5.18.1      piconv.1             tail.1
case.1                   iconservicesagent.1     perl5004delta5.16.1             perlhurd.1          piconv5.16.1         talk.1
cat.1                    iconservicesd.1         perl5004delta5.18.1             perlhurd5.16.1      piconv5.18.1         tar.1
cc_fips_test.1           iconset.1               perl5005delta.1                 perlhurd5.18.1      pidpersec.d.1m       taskinfo.1
cd.1                     icontool.1              perl5005delta5.16.1             perlintern.1        pkcs12.1ssl          tbl.1
cdcontexttool.1          iconutil.1              perl5005delta5.18.1             perlintern5.16.1    pkcs7.1ssl           tccutil.1
cdinteracttool.1         iconv.1                 perl5100delta.1                 perlintern5.18.1    pkcs8.1ssl           tclsh.1tcl
cdknowledgetool.1        id.1                    perl5100delta5.16.1             perlinterp.1        pkgbuild.1           tclsh8.4.1tcl
certtool.1               idlj.1                  perl5100delta5.18.1             perlinterp5.16.1    pkgutil.1            tclsh8.5.1tcl
cfilutil.1               if.1                    perl5101delta.1                 perlinterp5.18.1    pkill.1              tcpdump.1
chdir.1                  imptrace.1              perl5101delta5.16.1             perlintro.1         pl.1                 tcsh.1
checkLocalKDC.1          indxbib.1               perl5101delta5.18.1             perlintro5.16.1     pl2pm.1              tee.1
```

```
checknr.1                  info.1                 perl5120delta.1          perlintro5.18.1        pl2pm5.16.1            telltc.1
chflags.1                  infocmp.1m             perl5120delta5.16.1      perliol.1              pl2pm5.18.1            telnet.1
chfn.1                     infokey.1              perl5120delta5.18.1      perliol5.16.1          plockstat.1           test.1
chgrp.1                    infotocap.1m           perl5121delta.1          perliol5.18.1          plutil.1              texi2dvi.1
chmod.1                    install-info.1         perl5121delta5.16.1      perlipc.1              pmset.1               texindex.1
chpass.1                   install.1              perl5121delta5.18.1      perlipc5.16.1          pod2html.1            textutil.1
chsh.1                     instmodsh.1            perl5122delta.1          perlipc5.18.1          pod2html5.16.1        tfmtodit.1
ciphers.1ssl               instmodsh5.16.1        perl5122delta5.16.1      perlirix.1             pod2html5.18.1        tftp.1
cksum.1                    instmodsh5.18.1        perl5122delta5.18.1      perlirix5.16.1         pod2latex.1           then.1
clear.1                    intro.1                perl5123delta.1          perlirix5.18.1         pod2latex5.16.1       tic.1m
cmp.1                      iofile.d.1m            perl5123delta5.16.1      perlivp.1              pod2latex5.18.1       tidy.1
codesign.1                 iofileb.d.1m           perl5123delta5.18.1      perlivp5.16.1          pod2man.1             tiff2icns.1
col.1                      iopattern.1m           perl5124delta.1          perlivp5.18.1          pod2man5.16.1         tiffutil.1
colcrt.1                   iopending.1m           perl5124delta5.16.1      perljp.1               pod2man5.18.1         time.1
colldef.1                  iosnoop.1m             perl5124delta5.18.1      perljp5.16.1           pod2readme.1          timerfires.1
colrm.1                    iotop.1m               perl5125delta.1          perljp5.18.1           pod2readme5.16.1      times.1
column.1                   ip2cc.1                perl5125delta5.18.1      perlko.1               pod2readme5.18.1      tkcon.1
comm.1                     ip2cc5.16.1            perl5140delta.1          perlko5.16.1           pod2text.1            tkmib.1
command.1                  ip2cc5.18.1            perl5140delta5.16.1      perlko5.18.1           pod2text5.16.1        tkpp.1
complete.1                 ipcrm.1                perl5140delta5.18.1      perllexwarn.1          pod2text5.18.1        tkpp5.16.1
compress.1                 ipcs.1                 perl5141delta.1          perllexwarn5.16.1      pod2usage.1           tkpp5.18.1
config_data.1              ipmitool.1             perl5141delta5.16.1      perllexwarn5.18.1      pod2usage5.16.1       tnameserv.1
config_data5.16.1          ippfind.1.gz           perl5141delta5.18.1      perllinux.1            pod2usage5.18.1       toe.1m
config_data5.18.1          ipptool.1.gz           perl5142delta.1          perllinux5.16.1        podchecker.1          top.1
configureLocalKDC.1        irb.1                  perl5142delta5.16.1      perllinux5.18.1        podchecker5.16.1      tops.1
continue.1                 jar.1                  perl5142delta5.18.1      perllocale.1           podchecker5.18.1      topsyscall.1m
corelist.1                 jarsigner.1            perl5143delta.1          perllocale5.16.1       podselect.1           topsysproc.1m
corelist5.16.1             java.1                 perl5143delta5.16.1      perllocale5.18.1       podselect5.16.1       touch.1
corelist5.18.1             java_home.1            perl5143delta5.18.1      perllol.1              podselect5.18.1       tput.1
cp.1                       javac.1                perl5144delta.1          perllol5.16.1          policytool.1          tr.1
cpan.1                     javadoc.1              perl5144delta5.18.1      perllol5.18.1          popd.1                trace.1
cpan2dist.1                javah.1                perl5160delta.1          perlmacos.1            post-grohtml.1        trap.1
cpan2dist5.16.1            javap.1                perl5160delta5.16.1      perlmacos5.16.1        postalias.1           traptoemail.1
cpan2dist5.18.1            javaws.1               perl5160delta5.18.1      perlmacos5.18.1        postcat.1             treereg.1
cpan5.16.1                 jconsole.1             perl5161delta.1          perlmacosx.1           postconf.1            treereg5.16.1
cpan5.18.1                 jdb.1                  perl5161delta5.16.1      perlmacosx5.16.1       postdrop.1            treereg5.18.1
cpanp.1                    jhat.1                 perl5161delta5.18.1      perlmacosx5.18.1       postfix.1             troff.1
cpanp5.16.1                jinfo.1                perl5162delta.1          perlmod.1              postkick.1            true.1
cpanp5.18.1                jmap.1                 perl5162delta5.16.1      perlmod5.16.1          postlock.1            tset.1
cpio.1                     jobid.1                perl5162delta5.18.1      perlmod5.18.1          postlog.1             tsort.1
cpuwalk.d.1m               jobs.1                 perl5163delta.1          perlmodinstall.1       postmap.1             tty.1
creatbyproc.d.1m           join.1                 perl5163delta5.16.1      perlmodinstall5.16.1   postmulti.1           type.1
createhomedir.1            jot.1                  perl5163delta5.18.1      perlmodinstall5.18.1   postqueue.1           ul.1
crl.1ssl                   jps.1                  perl5180delta.1          perlmodlib.1           postsuper.1           ulimit.1
crl2pkcs7.1ssl             jrunscript.1           perl5180delta5.18.1      perlmodlib5.16.1       powermetrics.1        umask.1
crlrefresh.1               jsadebugd.1            perl5181delta.1          perlmodlib5.18.1       pp.1                  unalias.1
crontab.1                  json_pp.1              perl5181delta5.18.1      perlmodstyle.1         pp5.16.1              uname.1
csdiagnose.1               json_pp5.16.1          perl5182delta.1          perlmodstyle5.16.1     pp5.18.1              uncomplete.1
csgather.1                 json_pp5.18.1          perl5182delta5.18.1      perlmodstyle5.18.1     ppdc.1.gz             uncompress.1
csh.1                      jstack.1               perl561delta.1           perlmpeix.1            ppdhtml.1.gz          unexpand.1
csplit.1                   jstat.1                perl561delta5.16.1       perlmpeix5.16.1        ppdi.1.gz             unhash.1
csreq.1                    jstatd.1               perl561delta5.18.1       perlmroapi.1           ppdmerge.1.gz         uniq.1
cu.1                       kcc.1                  perl56delta.1            perlmroapi5.16.1       ppdpo.1.gz            units.1
cups-config.1.gz           kdestroy.1             perl56delta5.16.1        perlmroapi5.18.1       pr.1                  unlimit.1
cups.1.gz                  keytool.1              perl56delta5.18.1        perlnetware.1          praudit.1             unlink.1
cupstestdsc.1.gz           kgetcred.1             perl581delta.1           perlnetware5.16.1      pre-grohtml.1         unpack200.1
cupstestppd.1.gz           kill.1                 perl581delta5.16.1       perlnetware5.18.1      priclass.d.1m         unset.1
curl-config.1              kill.d.1m              perl581delta5.18.1       perlnewmod.1           pridist.d.1m          unsetenv.1
curl.1                     killall.1              perl582delta.1           perlnewmod5.16.1       printenv.1            until.1
cut.1                      kinit.1                perl582delta5.16.1       perlnewmod5.18.1       printf.1              unvis.1
cvaffinity.1               klist.1                perl582delta5.18.1       perlnumber.1           procsystime.1m        unzip.1
cvcp.1                     kpasswd.1              perl583delta.1           perlnumber5.16.1       productbuild.1        unzipsfx.1
cvmkdir.1                  kpconfig.1             perl583delta5.16.1       perlnumber5.18.1       productsign.1         update_dyld_shared_cache.1
cvmkfile.1                 kpfile.1               perl583delta5.18.1       perlobj.1              profiles.1            uptime.1
cvversions.1               kptop.1                perl584delta.1           perlobj5.16.1          prove.1               users.1
dappprof.1m                krb5-config.1          perl584delta5.16.1       perlobj5.18.1          prove5.16.1           uuconv.1
dapptrace.1m               ksh.1                  perl584delta5.18.1       perlootut.1            prove5.18.1           uucp.1
darwinup.1                 kswitch.1              perl585delta.1           perlootut5.16.1        ps.1                  uudecode.1
date.1                     lam.1                  perl585delta5.16.1       perlootut5.18.1        psed.1                uuencode.1
db_archive.1               languagesetup.1        perl585delta5.18.1       perlop.1               psed5.16.1            uuidgen.1
db_checkpoint.1            last.1                 perl586delta.1           perlop5.16.1           psed5.18.1            uulog.1
db_deadlock.1              lastcomm.1             perl586delta5.16.1       perlop5.18.1           pstopdf.1             uuname.1
db_dump.1                  lastwords.1m           perl586delta5.18.1       perlopenbsd.1          pstruct.1             uupick.1
db_load.1                  latency.1              perl587delta.1           perlopenbsd5.16.1      pstruct5.16.1         uustat.1
db_printlog.1              launchctl.1            perl587delta5.16.1       perlopenbsd5.18.1      pstruct5.18.1         uuto.1
db_recover.1               launchservicesd.1      perl587delta5.18.1       perlopentut.1          ptar.1                uux.1
db_stat.1                  layerutil.1            perl588delta.1           perlopentut5.16.1      ptar5.16.1            verify.1ssl
db_upgrade.1               ldapadd.1              perl588delta5.16.1       perlopentut5.18.1      ptar5.18.1            version.1ssl
db_verify.1                ldapcompare.1          perl588delta5.18.1       perlos2.1              ptardiff.1            vi.1
dbicadmin.1                ldapdelete.1           perl589delta.1           perlos25.16.1          ptardiff5.16.1        view.1
dbicadmin5.16.1            ldapexop.1             perl589delta5.16.1       perlos25.18.1          ptardiff5.18.1        vim.1
dbicadmin5.18.1            ldapmodify.1           perl589delta5.18.1       perlos390.1            ptargrep.1            vimdiff.1
dbilogstrip.1              ldapmodrdn.1           perl58delta.1            perlos3905.16.1        ptargrep5.16.1        vimtutor.1
dbilogstrip5.16.1          ldappasswd.1           perl58delta5.16.1        perlos3905.18.1        ptargrep5.18.1        vis.1
dbilogstrip5.18.1          ldapsearch.1           perl58delta5.18.1        perlos400.1            pubsub.1              vm_stat.1
dbiprof.1                  ldapurl.1              perlaix.1                perlos4005.16.1        pushd.1               vmmap.1
dbiprof5.16.1              ldapwhoami.1           perlaix5.16.1            perlos4005.18.1        pwd.1                 vmmap32.1
dbiprof5.18.1              leaks.1                perlaix5.18.1            perlpacktut.1          pwhich.1              w.1
dbiproxy.1                 leaks32.1              perlamiga.1              perlpacktut5.16.1      pwhich5.16.1          wait.1
dbiproxy5.16.1             leave.1                perlamiga5.16.1          perlpacktut5.18.1      pwhich5.18.1          wait4path.1
dbiproxy5.18.1             less.1.gz              perlamiga5.18.1          perlperf.1             pydoc.1               wall.1
dbmmanage.1                lessecho.1.gz          perlapi.1                perlperf5.16.1         pydoc2.6.1            wc.1
dc.1                       libnetcfg.1            perlapi5.16.1            perlperf5.18.1         pydoc2.7.1            weblatency.d.1m
dd.1                       libnetcfg5.16.1        perlapi5.18.1            perlplan9.1            python.1              what.1
debinhex.pl.1              libnetcfg5.18.1        perlapio.1               perlplan95.16.1        python2.6.1           whatis.1
debinhex.pl5.18.1          limit.1                perlapio5.16.1           perlplan95.18.1        python2.7.1           where.1
default.1                  link.1                 perlapio5.18.1           perlpod.1              pythonw.1             whereis.1
defaults.1                 lkbib.1                perlartistic.1           perlpod5.16.1          pythonw2.6.1          which.1
df.1                       ln.1                   perlartistic5.16.1       perlpod5.18.1          pythonw2.7.1          while.1
dhparam.1ssl               loads.d.1m             perlartistic5.18.1       perlpodspec.1          qc2movie.1            who.1
diff.1                     locale.1               perlbeos.1               perlpodspec5.16.1      q1manage.1            whoami.1
diff3.1                    localedef.1            perlbeos5.16.1           perlpodspec5.18.1      qtdefaults.1          whois.1
diffstat.1                 locate.1               perlbook.1               perlpodstyle.1         qtmodernizer.1        wish.1tcl
dig.1                      locationCLI.1          perlbook5.16.1           perlpodstyle5.16.1     quota.1               wish8.4.1tcl
dirname.1                  lockstat.1             perlbook5.18.1           perlpodstyle5.18.1     rake.1                wish8.5.1tcl
dirs.1                     log.1                  perlboot.1               perlpolicy.1           rand.1ssl             write.1
diskhits.1m                logger.1               perlboot5.16.1           perlpolicy5.16.1       rcp.1                 wsgen.1
dispqlen.d.1m              login.1                perlboot5.18.1           perlpolicy5.18.1       read.1                wsimport.1
ditto.1                    logname.1              perlbot.1                perlport.1             readlink.1            x509.1ssl
dns-sd.1                   logout.1               perlbot5.16.1            perlport5.16.1         readonly.1            xar.1
do.1                       logresolve.1           perlbot5.18.1            perlport5.18.1         red.1                 xargs.1
domainname.1               look.1                 perlbs2000.1             perlpragma.1           refer.1               xattr.1
done.1                     lookbib.1              perlbs20005.16.1         perlpragma5.16.1       rehash.1              xcode-select.1
dot_clean.1                lookupd.1              perlbs20005.18.1         perlpragma5.18.1       repeat.1              xgettext.pl.1
drutil.1                   lp.1.gz                perlbug.1                perlqnx.1              req.1ssl              xgettext.pl5.16.1
dsa.1ssl                   lpoptions.1.gz         perlbug5.16.1            perlqnx5.16.1          reset.1               xgettext.pl5.18.1
dsaparam.1ssl              lpq.1.gz               perlbug5.18.1            perlqnx5.18.1          return.1              xip.1
dscacheutil.1              lpr.1.gz               perlcall.1               perlre.1               rev.1                 xjc.1
dscl.1                     lprm.1.gz              perlcall5.16.1           perlre5.16.1           ri.1                  xml2-config.1
dsconfigldap.1             lpstat.1.gz            perlcall5.18.1           perlre5.18.1           rlogin.1              xmlcatalog.1
dsexport.1                 ls-F.1                 perlce.1                 perlreapi.1            rm.1                  xmllint.1
dsimport.1                 ls.1                   perlce5.16.1             perlreapi5.16.1        rmdir.1               xsltproc.1
```

```
dsmemberutil.1          lskq.1                  perlce5.18.1                perlreapi5.18.1         rmic.1              xsubpp.1
dtrace.1                lsm.1                   perlcheat.1                 perlrebackslash.1       rmid.1              xsubpp5.16.1
dtruss.1m               lsvfs.1                 perlcheat5.16.1             perlrebackslash5.16.1   rmiregistry.1       xsubpp5.18.1
du.1                    lwp-download.1          perlcheat5.18.1             perlrebackslash5.18.1   rs.1                xxd.1
dyld.1                  lwp-download5.16.1      perlclib.1                  perlrecharclass.1       rsa.1ssl            yes.1
ec.1ssl                 lwp-download5.18.1      perlclib5.16.1              perlrecharclass5.16.1   rsautl.1ssl         ypcat.1
echo.1                  lwp-dump.1              perlclib5.18.1              perlrecharclass5.18.1   rsh.1               ypmatch.1
echotc.1                lwp-dump5.16.1          perlcn.1                    perlref.1               rsync.1.gz          ypwhich.1
ecparam.1ssl            lwp-dump5.18.1          perlcn5.16.1                perlref5.16.1           ruby.1              zcat.1
ed.1                    lwp-mirror.1            perlcn5.18.1                perlref5.18.1           runocc.d.1m         zcmp.1
efax.1                  lwp-mirror5.16.1        perlcommunity.1             perlreftut.1            ruptime.1           zdiff.1
efix.1                  lwp-mirror5.18.1        perlcommunity5.16.1         perlreftut5.16.1        rview.1             zegrep.1
egrep.1                 lwp-request.1           perlcommunity5.18.1         perlreftut5.18.1        rvim.1              zfgrep.1
elif.1                  lwp-request5.16.1       perlcygwin.1                perlreguts.1            rwbypid.d.1m        zforce.1
else.1                  lwp-request5.18.1       perlcygwin5.16.1            perlreguts5.16.1        rwbytype.d.1m       zgrep.1
emacs-undumped.1        macbinary.1             perlcygwin5.18.1            perlreguts5.18.1        rwho.1              zip.1
emacs.1                 macerror.1              perldata.1                  perlrepository.1        rwsnoop.1m          zipcloak.1
emacsclient.1           macerror5.16.1          perldata5.16.1             perlrepository5.18.1    s2p.1               zipdetails.1
enc.1ssl                macerror5.18.1          perldata5.18.1             perlrequick.1           s2p5.16.1           zipdetails5.16.1
enc2xs.1                machine.1               perldbmfilter.1            perlrequick5.16.1       s2p5.18.1           zipdetails5.18.1
enc2xs5.16.1            mail.1                  perldbmfilter5.16.1        perlrequick5.18.1       s_client.1ssl       zipgrep.1
enc2xs5.18.1            mailq.1                 perldbmfilter5.18.1        perlreref.1             s_server.1ssl       zipinfo.1
encode_keychange.1      mailx.1                 perldebguts.1              perlreref5.16.1         s_time.1ssl         zipnote.1
end.1                   makeinfo.1              perldebguts5.16.1          perlreref5.18.1         safaridriver.1      zipsplit.1
endif.1                 malloc_history.1        perldebguts5.18.1          perlretut.1             sample.1            zless.1
endsw.1                 malloc_history32.1      perldebtut.1               perlretut5.16.1         sampleproc.1m       zmore.1
env.1                   man.1                   perldebtut5.16.1           perlretut5.18.1         sandbox-exec.1      znew.1
eqn.1                   manpath.1               perldebtut5.18.1           perlriscos.1            sar.1               zprint.1
eqn2graph.1             matchTool.1             perldebug.1                perlriscos5.16.1        say.1               zsh.1
erb.1                   mcxalr.1                perldebug5.16.1            perlriscos5.18.1        sc_usage.1          zshall.1
errinfo.1m              mcxquery.1              perldebug5.18.1            perlrun.1               scandeps.pl.1       zshbuiltins.1
errstr.1ssl             mcxrefresh.1            perldelta.1                perlrun5.16.1           scandeps.pl5.16.1   zshcompctl.1
esac.1                  md5.1                   perldelta5.16.1            perlrun5.18.1           scandeps.pl5.18.1   zshcompsys.1
etags.1                 mdfind.1                perldelta5.18.1            perlsec.1               sched.1             zshcompwid.1
eval.1                  mdimport.1              perldgux.1                 perlsec5.16.1           schemagen.1         zshcontrib.1
evim.1                  mdimport32.1            perldgux5.16.1             perlsec5.18.1           scp.1               zshexpn.1
ex.1                    mdls.1                  perldgux5.18.1             perlsolaris.1           screen.1            zshmisc.1
exec.1                  mdmclient.1             perldiag.1                 perlsolaris5.16.1       screencapture.1     zshmodules.1
execsnoop.1m            mdutil.1                perldiag5.16.1            perlsolaris5.18.1       script.1            zshoptions.1
exit.1                  memberd.1               perldiag5.18.1            perlsource.1            sdiff.1             zshparam.1
expand.1                memory_pressure.1       perldoc.1                 perlsource5.16.1        sdx.1               zshroadmap.1
expect.1                mesg.1                  perldoc5.16.1             perlsource5.18.1        security.1          zshtcpsys.1
export.1                mib2c-update.1          perldoc5.18.1             perlstyle.1             securityd.1         zshzftpsys.1
expr.1                  mib2c.1                 perldos.1                 perlstyle5.16.1         sed.1               zshzle.1
extcheck.1              migrateLocalKDC.1       perldos5.16.1             perlstyle5.18.1         seeksize.d.1m
eyapp.1                 mkdir.1                 perldos5.18.1             perlsub.1               sendmail.1
eyapp5.16.1             mkfifo.1                perldsc.1                 perlsub5.16.1           seq.1
[wansongHome@wansong:/usr/share/man/man1]$ cat bash
bash.1     bashbug.1
[wansongHome@wansong:/usr/share/man/man1]$ cat bash
bash.1     bashbug.1
[wansongHome@wansong:/usr/share/man/man1]$ cat bash
bash.1     bashbug.1
[wansongHome@wansong:/usr/share/man/man1]$ cat bash.1
.\"
.\" MAN PAGE COMMENTS to
.\"
.\"     Chet Ramey
.\"     Information Network Services
.\"     Case Western Reserve University
.\"     chet@po.cwru.edu
.\"
.\"     Last Change: Thu Sep 28 10:25:59 EDT 2006
.\"
.\" bash_builtins, strip all but Built-Ins section
.if \n(zZ=1 .ig zZ
.if \n(zY=1 .ig zY
.TH BASH 1 "2006 September 28" "GNU Bash-3.2"
.\"
.\" There's some problem with having a `@'
.\" in a tagged paragraph with the BSD man macros.
.\" It has to do with `@' appearing in the }1 macro.
.\" This is a problem on 4.3 BSD and Ultrix, but Sun
.\" appears to have fixed it.
.\" If you're seeing the characters
.\" `@u-3p' appearing before the lines reading
.\" `possible-hostname-completions
.\" and `complete-hostname' down in READLINE,
.\" then uncomment this redefinition.
.\"
.de }1
.ds ]X \&\\*(]B\\
.nr )E 0
.if !"\\$1"" .nr )I \\$1n
.}f
.ll \\n(LLu
.in \\n()Ru+\\n(INu+\\n()Iu
.ti \\n(INu
.ie !\\n()Iu+\\n()Ru-\w\\*(]Xu-3p \{\\*(]X
.br\}
.el \\*(]X\h|\\n()Iu+\\n()Ru\c
.}f
..
.\"
.\" File Name macro.  This used to be `.PN', for Path Name,
.\" but Sun doesn't seem to like that very much.
.\"
.de FN
\fI\|\\$1\|\fP
..
.SH NAME
bash \- GNU Bourne-Again SHell
.SH SYNOPSIS
.B bash
[options]
[file]
.SH COPYRIGHT
.if n Bash is Copyright (C) 1989-2005 by the Free Software Foundation, Inc.
.if t Bash is Copyright \(co 1989-2005 by the Free Software Foundation, Inc.
.SH DESCRIPTION
.B Bash
is an \fBsh\fR-compatible command language interpreter that
executes commands read from the standard input or from a file.
.B Bash
also incorporates useful features from the \fIKorn\fP and \fIC\fP
shells (\fBksh\fP and \fBcsh\fP).
.PP
.B Bash
is intended to be a conformant implementation of the
Shell and Utilities portion of the IEEE POSIX specification
(IEEE Standard 1003.1).
.B Bash
```

```
can be configured to be POSIX-conformant by default.
.SH OPTIONS
In addition to the single-character shell options documented in the
description of the \fBset\fR builtin command, \fBbash\fR
interprets the following options when it is invoked:
.PP
.PD 0
.TP 10
.BI \-c "\| string\^"
If the
.B \-c
option is present, then commands are read from
.IR string .
If there are arguments after the
.IR string ,
they are assigned to the positional parameters, starting with
.BR $0 .
.TP
.B \-i
If the
.B \-i
option is present, the shell is
.IR interactive .
.TP
.B \-l
Make
.B bash
act as if it had been invoked as a login shell (see
.SM
.B INVOCATION
below).
.TP
.B \-r
If the
.B \-r
option is present, the shell becomes
.I restricted
(see
.SM
.B "RESTRICTED SHELL"
below).
.TP
.B \-s
If the
.B \-s
option is present, or if no arguments remain after option
processing, then commands are read from the standard input.
This option allows the positional parameters to be set
when invoking an interactive shell.
.TP
.B \-D
A list of all double-quoted strings preceded by \fB$\fP
is printed on the standard output.
These are the strings that
are subject to language translation when the current locale
is not \fBC\fP or \fBPOSIX\fP.
This implies the \fB\-n\fP option; no commands will be executed.
.TP
.B [\-+]O [\fIshopt_option\fP]
\fIshopt_option\fP is one of the shell options accepted by the
\fBshopt\fP builtin (see
.SM
.B SHELL BUILTIN COMMANDS
below).
If \fIshopt_option\fP is present, \fB\-O\fP sets the value of that option;
\fB+O\fP unsets it.
If \fIshopt_option\fP is not supplied, the names and values of the shell
options accepted by \fBshopt\fP are printed on the standard output.
If the invocation option is \fB+O\fP, the output is displayed in a format
that may be reused as input.
.TP
.B \-\-
A
.B \-\-
signals the end of options and disables further option processing.
Any arguments after the
.B \-\-
are treated as filenames and arguments.  An argument of
.B \-
is equivalent to \fB\-\-\fP.
.PD
.PP
.B Bash
also interprets a number of multi-character options.
These options must appear on the command line before the
single-character options to be recognized.
.PP
.PD 0
.TP
.B \-\-debugger
Arrange for the debugger profile to be executed before the shell
starts.
Turns on extended debugging mode (see the description of the
.B extdebug
option to the
.B shopt
builtin below)
and shell function tracing (see the description of the
\fB\-o functrace\fP option to the
.B set
builtin below).
.TP
.B \-\-dump\-po\-strings
Equivalent to \fB\-D\fP, but the output is in the GNU \fIgettext\fP
\fBpo\fP (portable object) file format.
.TP
.B \-\-dump\-strings
Equivalent to \fB\-D\fP.
.TP
.B \-\-help
Display a usage message on standard output and exit successfully.
.TP
\fB\-\-init\-file\fP \fIfile\fP
.PD 0
.TP
\fB\-\-rcfile\fP \fIfile\fP
.PD
Execute commands from
.I file
instead of the standard personal initialization file
.I ~/.bashrc
if the shell is interactive (see
```

.SM
.B INVOCATION
below).
.TP
.B \-\-login
Equivalent to \fB\-l\fP.
.TP
.B \-\-noediting
Do not use the GNU
.B readline
library to read command lines when the shell is interactive.
.TP
.B \-\-noprofile
Do not read either the system-wide startup file
.FN /etc/profile
or any of the personal initialization files
.IR ~/.bash_profile ,
.IR ~/.bash_login ,
or
.IR ~/.profile .
By default,
.B bash
reads these files when it is invoked as a login shell (see
.SM
.B INVOCATION
below).
.TP
.B \-\-norc
Do not read and execute the personal initialization file
.I ~/.bashrc
if the shell is interactive.
This option is on by default if the shell is invoked as
.BR sh .
.TP
.B \-\-posix
Change the behavior of \fBbash\fP where the default operation differs
from the POSIX standard to match the standard (\fIposix mode\fP).
.TP
.B \-\-restricted
The shell becomes restricted (see
.SM
.B "RESTRICTED SHELL"
below).
.TP
.B \-\-verbose
Equivalent to \fB\-v\fP.
.TP
.B \-\-version
Show version information for this instance of
.B bash
on the standard output and exit successfully.
.PD
.SH ARGUMENTS
If arguments remain after option processing, and neither the
.B \-c
nor the
.B \-s
option has been supplied, the first argument is assumed to
be the name of a file containing shell commands.
If
.B bash
is invoked in this fashion,
.B $0
is set to the name of the file, and the positional parameters
are set to the remaining arguments.
.B Bash
reads and executes commands from this file, then exits.
\fBBash\fP's exit status is the exit status of the last command
executed in the script.
If no commands are executed, the exit status is 0.
An attempt is first made to open the file in the current directory, and,
if no file is found, then the shell searches the directories in
.SM
.B PATH
for the script.
.SH INVOCATION
A \fIlogin shell\fP is one whose first character of argument zero is a
.BR \- ,
or one started with the
.B \-\-login
option.
.PP
An \fIinteractive\fP shell is one started without non-option arguments
and without the
.B \-c
option
whose standard input and error are
both connected to terminals (as determined by
.IR isatty (3)),
or one started with the
.B \-i
option.
.SM
.B PS1
is set and
.B $\-
includes
.B i
if
.B bash
is interactive,
allowing a shell script or a startup file to test this state.
.PP
The following paragraphs describe how
.B bash
executes its startup files.
If any of the files exist but cannot be read,
.B bash
reports an error.
Tildes are expanded in file names as described below under
.B "Tilde Expansion"
in the
.SM
.B EXPANSION
section.
.PP
When
.B bash
is invoked as an interactive login shell, or as a non-interactive shell
with the \fB\-\-login\fP option, it first reads and
executes commands from the file \fI/etc/profile\fP, if that
file exists.

After reading that file, it looks for \fI~/.bash_profile\fP,
\fI~/.bash_login\fP, and \fI~/.profile\fP, in that order, and reads
and executes commands from the first one that exists and is readable.
The
.B \-\-noprofile
option may be used when the shell is started to inhibit this behavior.
.PP
When a login shell exits,
.B bash
reads and executes commands from the file \fI~/.bash_logout\fP, if it
exists.
.PP
When an interactive shell that is not a login shell is started,
.B bash
reads and executes commands from \fI~/.bashrc\fP, if that file exists.
This may be inhibited by using the
.B \-\-norc
option.
The \fB\-\-rcfile\fP \fIfile\fP option will force
.B bash
to read and execute commands from \fIfile\fP instead of \fI~/.bashrc\fP.
.PP
When
.B bash
is started non-interactively, to run a shell script, for example, it
looks for the variable
.SM
.B BASH_ENV
in the environment, expands its value if it appears there, and uses the
expanded value as the name of a file to read and execute.
.B Bash
behaves as if the following command were executed:
.sp .5
.RS
.if t \f(CWif [\-n "$BASH_ENV"]; then . "$BASH_ENV"; fi\fP
.if n if [\-n "$BASH_ENV"]; then . "$BASH_ENV"; fi
.RE
.sp .5
but the value of the
.SM
.B PATH
variable is not used to search for the file name.
.PP
If
.B bash
is invoked with the name
.BR sh ,
it tries to mimic the startup behavior of historical versions of
.B sh
as closely as possible,
while conforming to the POSIX standard as well.
When invoked as an interactive login shell, or a non-interactive
shell with the \fB\-\-login\fP option, it first attempts to
read and execute commands from
.I /etc/profile
and
.IR ~/.profile ,
in that order.
The
.B \-\-noprofile
option may be used to inhibit this behavior.
When invoked as an interactive shell with the name
.BR sh ,
.B bash
looks for the variable
.SM
.BR ENV ,
expands its value if it is defined, and uses the
expanded value as the name of a file to read and execute.
Since a shell invoked as
.B sh
does not attempt to read and execute commands from any other startup
files, the
.B \-\-rcfile
option has no effect.
A non-interactive shell invoked with the name
.B sh
does not attempt to read any other startup files.
When invoked as
.BR sh ,
.B bash
enters
.I posix
mode after the startup files are read.
.PP
When
.B bash
is started in
.I posix
mode, as with the
.B \-\-posix
command line option, it follows the POSIX standard for startup files.
In this mode, interactive shells expand the
.SM
.B ENV
variable and commands are read and executed from the file
whose name is the expanded value.
No other startup files are read.
.PP
.B Bash
attempts to determine when it is being run by the remote shell
daemon, usually \fIrshd\fP.
If
.B bash
determines it is being run by \fIrshd\fP, it reads and executes
commands from \fI~/.bashrc\fP, if that file exists and is readable.
It will not do this if invoked as \fBsh\fP.
The
.B \-\-norc
option may be used to inhibit this behavior, and the
.B \-\-rcfile
option may be used to force another file to be read, but
\fIrshd\fP does not generally invoke the shell with those options
or allow them to be specified.
.PP
If the shell is started with the effective user (group) id not equal to the
real user (group) id, and the \fB\-p\fP option is not supplied, no startup
files are read, shell functions are not inherited from the environment, the
.SM
.B SHELLOPTS
variable, if it appears in the environment, is ignored,
and the effective user id is set to the real user id.

If the \fB\-p\fP option is supplied at invocation, the startup behavior is
the same, but the effective user id is not reset.
.SH DEFINITIONS
.PP
The following definitions are used throughout the rest of this
document.
.PD 0
.TP
.B blank
A space or tab.
.TP
.B word
A sequence of characters considered as a single unit by the shell.
Also known as a
.BR token .
.TP
.B name
A
.I word
consisting only of alphanumeric characters and underscores, and
beginning with an alphabetic character or an underscore. Also
referred to as an
.BR identifier .
.TP
.B metacharacter
A character that, when unquoted, separates words. One of the following:
.br
.RS
.PP
.if t \fB| &   ;   (   )   <   >   space   tab\fP
.if n \fB| & ; ( ) < > space tab\fP
.RE
.PP
.TP
.B control operator
A \fItoken\fP that performs a control function. It is one of the following
symbols:
.RS
.PP
.if t \fB\(bv\(bv   &   &&   ;   ;;   (   ) | <newline>\fP
.if n \fB|| & && ; ;; ( ) | <newline>\fP
.RE
.PD
.SH "RESERVED WORDS"
\fIReserved words\fP are words that have a special meaning to the shell.
The following words are recognized as reserved when unquoted and either
the first word of a simple command (see
.SM
.B SHELL GRAMMAR
below) or the third word of a
.B case
or
.B for
command:
.if t .RS
.PP
.B
.if n ! case do done elif else esac fi for function if in select then until while { } time [[ ]]
.if t !   case   do   done   elif   else   esac   fi   for   function   if   in   select   then   until   while   {   }   time   [[   ]]
.if t .RE
.SH "SHELL GRAMMAR"
.SS Simple Commands
.PP
A \fIsimple command\fP is a sequence of optional variable assignments
followed by \fBblank\fP-separated words and redirections, and
terminated by a \fIcontrol operator\fP. The first word
specifies the command to be executed, and is passed as argument zero.
The remaining words are passed as arguments to the invoked command.
.PP
The return value of a \fIsimple command\fP is its exit status, or
128+\fIn\^\fP if the command is terminated by signal
.IR n .
.SS Pipelines
.PP
A \fIpipeline\fP is a sequence of one or more commands separated by
the character
.BR | .
The format for a pipeline is:
.RS
.PP
[\fBtime\fP [\fB\-p\fP]] [!] \fIcommand\fP [\fB|\fP \fIcommand2\fP ...]
.RE
.PP
The standard output of
.I command
is connected via a pipe to the standard input of
.IR command2 .
This connection is performed before any redirections specified by the
command (see
.SM
.B REDIRECTION
below).
.PP
The return status of a pipeline is the exit status of the last
command, unless the \fBpipefail\fP option is enabled.
If \fBpipefail\fP is enabled, the pipeline's return status is the
value of the last (rightmost) command to exit with a non-zero status,
or zero if all commands exit successfully.
If the reserved word
.B !
precedes a pipeline, the exit status of that pipeline is the logical
negation of the exit status as described above.
The shell waits for all commands in the pipeline to
terminate before returning a value.
.PP
If the
.B time
reserved word precedes a pipeline, the elapsed as well as user and
system time consumed by its execution are reported when the pipeline
terminates.
The \fB\-p\fP option changes the output format to that specified by POSIX.
The
.SM
.B TIMEFORMAT
variable may be set to a format string that specifies how the timing
information should be displayed; see the description of
.SM
.B TIMEFORMAT
under
.B "Shell Variables"
below.
.PP

Each command in a pipeline is executed as a separate process (i.e., in a
subshell).
.SS Lists
.PP
A \fIlist\fP is a sequence of one or more pipelines separated by one
of the operators
.BR ; ,
.BR & ,
.BR && ,
or
.BR \(bv\(bv ,
and optionally terminated by one of
.BR ; ,
.BR & ,
or
.BR <newline> .
.PP
Of these list operators,
.B &&
and
.B \(bv\(bv
have equal precedence, followed by
.B ;
and
.BR &,
which have equal precedence.
.PP
A sequence of one or more newlines may appear in a \fIlist\fP instead
of a semicolon to delimit commands.
.PP
If a command is terminated by the control operator
.BR & ,
the shell executes the command in the \fIbackground\fP
in a subshell. The shell does not wait for the command to
finish, and the return status is 0. Commands separated by a
.B ;
are executed sequentially; the shell waits for each
command to terminate in turn. The return status is the
exit status of the last command executed.
.PP
The control operators
.B &&
and
.B \(bv\(bv
denote AND lists and OR lists, respectively.
An AND list has the form
.RS
.PP
\fIcommand1\fP \fB&&\fP \fIcommand2\fP
.RE
.PP
.I command2
is executed if, and only if,
.I command1
returns an exit status of zero.
.PP
An OR list has the form
.RS
.PP
\fIcommand1\fP \fB\(bv\(bv\fP \fIcommand2\fP
.PP
.RE
.PP
.I command2
is executed if and only if
.I command1
returns a non-zero exit status. The return status of
AND and OR lists is the exit status of the last command
executed in the list.
.SS Compound Commands
.PP
A \fIcompound command\fP is one of the following:
.TP
(\fIlist\fP)
\fIlist\fP is executed in a subshell environment (see
.SM
\fBCOMMAND EXECUTION ENVIRONMENT\fP
below).
Variable assignments and builtin
commands that affect the shell's environment do not remain in effect
after the command completes. The return status is the exit status of
\fIlist\fP.
.TP
{ \fIlist\fP; }
\fIlist\fP is simply executed in the current shell environment.
\fIlist\fP must be terminated with a newline or semicolon.
This is known as a \fIgroup command\fP.
The return status is the exit status of
\fIlist\fP.
Note that unlike the metacharacters \fB(\fP and \fB)\fP, \fB{\fP and
\fB}\fP are \fIreserved words\fP and must occur where a reserved
word is permitted to be recognized. Since they do not cause a word
break, they must be separated from \fIlist\fP by whitespace.
.TP
((\fIexpression\fP))
The \fIexpression\fP is evaluated according to the rules described
below under
.SM
.BR "ARITHMETIC EVALUATION" .
If the value of the expression is non-zero, the return status is 0;
otherwise the return status is 1. This is exactly equivalent to
\fBlet "\fIexpression\fP"\fR.
.TP
\fB[[\fP \fIexpression\fP \fB]]\fP
Return a status of 0 or 1 depending on the evaluation of
the conditional expression \fIexpression\fP.
Expressions are composed of the primaries described below under
.SM
.BR "CONDITIONAL EXPRESSIONS" .
Word splitting and pathname expansion are not performed on the words
between the \fB[[\fP and \fB]]\fP; tilde expansion, parameter and
variable expansion, arithmetic expansion, command substitution, process
substitution, and quote removal are performed.
Conditional operators such as \fB\-f\fP must be unquoted to be recognized
as primaries.
.if t .sp 0.5
.if n .sp 1
When the \fB==\fP and \fB!=\fP operators are used, the string to the
right of the operator is considered a pattern and matched according
to the rules described below under \fBPattern Matching\fP.
If the shell option
.B nocasematch

is enabled, the match is performed without regard to the case
of alphabetic characters.
The return value is 0 if the string matches (\fB==\fP) or does not match
(\fB!=\fP) the pattern, and 1 otherwise.
Any part of the pattern may be quoted to force it to be matched as a
string.
.if t .sp 0.5
.if n .sp 1
An additional binary operator, \fB=~\fP, is available, with the same
precedence as \fB==\fP and \fB!=\fP.
When it is used, the string to the right of the operator is considered
an extended regular expression and matched accordingly (as in \fIregex\fP(3)).
The return value is 0 if the string matches
the pattern, and 1 otherwise.
If the regular expression is syntactically incorrect, the conditional
expression's return value is 2.
If the shell option
.B nocasematch
is enabled, the match is performed without regard to the case
of alphabetic characters.
Substrings matched by parenthesized subexpressions within the regular
expression are saved in the array variable \fBBASH_REMATCH\fP.
The element of \fBBASH_REMATCH\fP with index 0 is the portion of the string
matching the entire regular expression.
The element of \fBBASH_REMATCH\fP with index \fIn\fP is the portion of the
string matching the \fIn\fPth parenthesized subexpression.
.if t .sp 0.5
.if n .sp 1
Expressions may be combined using the following operators, listed
in decreasing order of precedence:
.if t .sp 0.5
.if n .sp 1
.RS
.PD 0
.TP
.B (\fIexpression\fP)
Returns the value of \fIexpression\fP.
This may be used to override the normal precedence of operators.
.TP
.B ! \fIexpression\fP
True if
.I expression
is false.
.TP
\fIexpression1\fP \fB&&\fP \fIexpression2\fP
True if both
.I expression1
and
.I expression2
are true.
.TP
.if t \fIexpression1\fP \fB\(bv\(bv\fP \fIexpression2\fP
.if n \fIexpression1\fP \fB||\fP \fIexpression2\fP
True if either
.I expression1
or
.I expression2
is true.
.PD
.LP
The \fB&&\fP and
.if t \fB\(bv\(bv\fP
.if n \fB||\fP
operators do not evaluate \fIexpression2\fP if the value of
\fIexpression1\fP is sufficient to determine the return value of
the entire conditional expression.
.RE
.TP
\fBfor\fP \fIname\fP [\fBin\fP \fIword\fP] ; \fBdo\fP \fIlist\fP ; \fBdone\fP
The list of words following \fBin\fP is expanded, generating a list
of items.
The variable \fIname\fP is set to each element of this list
in turn, and \fIlist\fP is executed each time.
If the \fBin\fP \fIword\fP is omitted, the \fBfor\fP command executes
\fIlist\fP once for each positional parameter that is set (see
.SM
.B PARAMETERS
below).
The return status is the exit status of the last command that executes.
If the expansion of the items following \fBin\fP results in an empty
list, no commands are executed, and the return status is 0.
.TP
\fBfor\fP ((\fIexpr1\fP ; \fIexpr2\fP ; \fIexpr3\fP)) ; \fBdo\fP \fIlist\fP ; \fBdone\fP
First, the arithmetic expression \fIexpr1\fP is evaluated according
to the rules described below under
.SM
.BR "ARITHMETIC EVALUATION" .
The arithmetic expression \fIexpr2\fP is then evaluated repeatedly
until it evaluates to zero.
Each time \fIexpr2\fP evaluates to a non-zero value, \fIlist\fP is
executed and the arithmetic expression \fIexpr3\fP is evaluated.
If any expression is omitted, it behaves as if it evaluates to 1.
The return value is the exit status of the last command in \fIlist\fP
that is executed, or false if any of the expressions is invalid.
.TP
\fBselect\fP \fIname\fP [\fBin\fP \fIword\fP] ; \fBdo\fP \fIlist\fP ; \fBdone\fP
The list of words following \fBin\fP is expanded, generating a list
of items. The set of expanded words is printed on the standard
error, each preceded by a number. If the \fBin\fP
\fIword\fP is omitted, the positional parameters are printed (see
.SM
.B PARAMETERS
below). The
.B PS3
prompt is then displayed and a line read from the standard input.
If the line consists of a number corresponding to one of
the displayed words, then the value of
.I name
is set to that word. If the line is empty, the words and prompt
are displayed again. If EOF is read, the command completes. Any
other value read causes
.I name
to be set to null. The line read is saved in the variable
.BR REPLY .
The
.I list
is executed after each selection until a
.B break
command is executed.
The exit status of
.B select
is the exit status of the last command executed in

```
.IR list ,
or zero if no commands were executed.
.TP
\fBcase\fP \fIword\fP \fBin\fP [ [(] \fIpattern\fP [ \fB|\fP \fIpattern\fP ] \
... ) \fIlist\fP ;; ] ... \fBesac\fP
A \fBcase\fP command first expands \fIword\fP, and tries to match
it against each \fIpattern\fP in turn, using the same matching rules
as for pathname expansion (see
.B Pathname Expansion
below).
The \fIword\fP is expanded using tilde
expansion, parameter and variable expansion, arithmetic substitution,
command substitution, process substitution and quote removal.
Each \fIpattern\fP examined is expanded using tilde
expansion, parameter and variable expansion, arithmetic substitution,
command substitution, and process substitution.
If the shell option
.B nocasematch
is enabled, the match is performed without regard to the case
of alphabetic characters.
When a match is found, the
corresponding \fIlist\fP is executed.  After the first match, no
subsequent matches are attempted.  The exit status is zero if no
pattern matches.  Otherwise, it is the exit status of the
last command executed in \fIlist\fP.
.TP
\fBif\fP \fIlist\fP; \fBthen\fP \fIlist;\fP \
[ \fBelif\fP \fIlist\fP; \fBthen\fP \fIlist\fP; ] ... \
[ \fBelse\fP \fIlist\fP; ] \fBfi\fP
The
.B if
.I list
is executed.  If its exit status is zero, the
\fBthen\fP \fIlist\fP is executed.  Otherwise, each \fBelif\fP
\fIlist\fP is executed in turn, and if its exit status is zero,
the corresponding \fBthen\fP \fIlist\fP is executed and the
command completes.  Otherwise, the \fBelse\fP \fIlist\fP is
executed, if present.  The exit status is the exit status of the
last command executed, or zero if no condition tested true.
.TP
\fBwhile\fP \fIlist\fP; \fBdo\fP \fIlist\fP; \fBdone\fP
.PD 0
.TP
\fBuntil\fP \fIlist\fP; \fBdo\fP \fIlist\fP; \fBdone\fP
.PD
The \fBwhile\fP command continuously executes the \fBdo\fP
\fIlist\fP as long as the last command in \fIlist\fP returns
an exit status of zero.  The \fBuntil\fP command is identical
to the \fBwhile\fP command, except that the test is negated;
the
.B do
.I list
is executed as long as the last command in
.I list
returns a non-zero exit status.
The exit status of the \fBwhile\fP and \fBuntil\fP commands
is the exit status
of the last \fBdo\fP \fIlist\fP command executed, or zero if
none was executed.
.SS Shell Function Definitions
.PP
A shell function is an object that is called like a simple command and
executes a compound command with a new set of positional parameters.
Shell functions are declared as follows:
.TP
[ \fBfunction\fP ] \fIname\fP () \fIcompound\-command\fP [\fIredirection\fP]
This defines a function named \fIname\fP.
The reserved word \fBfunction\fP is optional.
If the \fBfunction\fP reserved word is supplied, the parentheses are optional.
The \fIbody\fP of the function is the compound command
.I compound\-command
(see \fBCompound Commands\fP above).
That command is usually a \fIlist\fP of commands between { and }, but
may be any command listed under \fBCompound Commands\fP above.
\fIcompound\-command\fP is executed whenever \fIname\fP is specified as the
name of a simple command.
Any redirections (see
.SM
.B REDIRECTION
below) specified when a function is defined are performed
when the function is executed.
The exit status of a function definition is zero unless a syntax error
occurs or a readonly function with the same name already exists.
When executed, the exit status of a function is the exit status of the
last command executed in the body.  (See
.SM
.B FUNCTIONS
below.)
.SH COMMENTS
In a non-interactive shell, or an interactive shell in which the
.B interactive_comments
option to the
.B shopt
builtin is enabled (see
.SM
.B "SHELL BUILTIN COMMANDS"
below), a word beginning with
.B #
causes that word and all remaining characters on that line to
be ignored.  An interactive shell without the
.B interactive_comments
option enabled does not allow comments.  The
.B interactive_comments
option is on by default in interactive shells.
.SH QUOTING
\fIQuoting\fP is used to remove the special meaning of certain
characters or words to the shell.  Quoting can be used to
disable special treatment for special characters, to prevent
reserved words from being recognized as such, and to prevent
parameter expansion.
.PP
Each of the \fImetacharacters\fP listed above under
.SM
.B DEFINITIONS
has special meaning to the shell and must be quoted if it is to
represent itself.
.PP
When the command history expansion facilities are being used
(see
.SM
.B HISTORY EXPANSION
below), the
```

\fIhistory expansion\fP character, usually \fB!\fP, must be quoted
to prevent history expansion.
.PP
There are three quoting mechanisms: the
.IR "escape character" ,
single quotes, and double quotes.
.PP
A non-quoted backslash (\fB\e\fP) is the
.IR "escape character" .
It preserves the literal value of the next character that follows,
with the exception of <newline>. If a \fB\e\fP<newline> pair
appears, and the backslash is not itself quoted, the \fB\e\fP<newline>
is treated as a line continuation (that is, it is removed from the
input stream and effectively ignored).
.PP
Enclosing characters in single quotes preserves the literal value
of each character within the quotes. A single quote may not occur
between single quotes, even when preceded by a backslash.
.PP
Enclosing characters in double quotes preserves the literal value
of all characters within the quotes, with the exception of
.BR $,
.BR ` ,
.BR \e ,
and, when history expansion is enabled,
.BR ! .
The characters
.B $
and
.B `
retain their special meaning within double quotes. The backslash
retains its special meaning only when followed by one of the following
characters:
.BR $,
.BR ` ,
\^\fB"\fP\^,
.BR \e ,
or
.BR <newline> .
A double quote may be quoted within double quotes by preceding it with
a backslash.
If enabled, history expansion will be performed unless an
.B !
appearing in double quotes is escaped using a backslash.
The backslash preceding the
.B !
is not removed.
.PP
The special parameters
.B *
and
.B @
have special meaning when in double
quotes (see
.SM
.B PARAMETERS
below).
.PP
Words of the form \fB$\fP\(aq\fIstring\fP\(aq are treated specially. The
word expands to \fIstring\fP, with backslash-escaped characters replaced
as specified by the ANSI C standard. Backslash escape sequences, if
present, are decoded as follows:
.RS
.PD 0
.TP
.B \ea
alert (bell)
.TP
.B \eb
backspace
.TP
.B \ee
an escape character
.TP
.B \ef
form feed
.TP
.B \en
new line
.TP
.B \er
carriage return
.TP
.B \et
horizontal tab
.TP
.B \ev
vertical tab
.TP
.B \e\e
backslash
.TP
.B \e\(aq
single quote
.TP
.B \e\fInnn\fP
the eight-bit character whose value is the octal value \fInnn\fP
(one to three digits)
.TP
.B \ex\fIHH\fP
the eight-bit character whose value is the hexadecimal value \fIHH\fP
(one or two hex digits)
.TP
.B \ec\fIx\fP
a control-\fIx\fP character
.PD
.RE
.LP
The expanded result is single-quoted, as if the dollar sign had
not been present.
.PP
A double-quoted string preceded by a dollar sign (\fB$\fP) will cause
the string to be translated according to the current locale.
If the current locale is \fBC\fP or \fBPOSIX\fP, the dollar sign
is ignored.
If the string is translated and replaced, the replacement is
double-quoted.
.SH PARAMETERS
A
.I parameter
is an entity that stores values.
It can be a

.IR name ,
a number, or one of the special characters listed below under
.BR "Special Parameters" .
A
.I variable
is a parameter denoted by a
.IR name .
A variable has a \fIvalue\fP and zero or more \fIattributes\fP.
Attributes are assigned using the
.B declare
builtin command (see
.B declare
below in
.SM
.BR "SHELL BUILTIN COMMANDS").
.PP
A parameter is set if it has been assigned a value. The null string is
a valid value. Once a variable is set, it may be unset only by using
the
.B unset
builtin command (see
.SM
.B SHELL BUILTIN COMMANDS
below).
.PP
A
.I variable
may be assigned to by a statement of the form
.RS
.PP
\fIname\fP=[\fIvalue\fP]
.RE
.PP
If
.I value
is not given, the variable is assigned the null string. All
.I values
undergo tilde expansion, parameter and variable expansion,
command substitution, arithmetic expansion, and quote
removal (see
.SM
.B EXPANSION
below). If the variable has its
.B integer
attribute set, then
.I value
is evaluated as an arithmetic expression even if the $((...)) expansion is
not used (see
.B "Arithmetic Expansion"
below).
Word splitting is not performed, with the exception
of \fB"$@"\fP as explained below under
.BR "Special Parameters" .
Pathname expansion is not performed.
Assignment statements may also appear as arguments to the
.BR alias ,
.BR declare ,
.BR typeset ,
.BR export ,
.BR readonly ,
and
.B local
builtin commands.
.PP
In the context where an assignment statement is assigning a value
to a shell variable or array index, the += operator can be used to
append to or add to the variable's previous value.
When += is applied to a variable for which the integer attribute has been
set, \fIvalue\fP is evaluated as an arithmetic expression and added to the
variable's current value, which is also evaluated.
When += is applied to an array variable using compound assignment (see
.B Arrays
below), the
variable's value is not unset (as it is when using =), and new values are
appended to the array beginning at one greater than the array's maximum index.
When applied to a string-valued variable, \fIvalue\fP is expanded and
appended to the variable's value.
.SS Positional Parameters
.PP
A
.I positional parameter
is a parameter denoted by one or more
digits, other than the single digit 0. Positional parameters are
assigned from the shell's arguments when it is invoked,
and may be reassigned using the
.B set
builtin command. Positional parameters may not be assigned to
with assignment statements. The positional parameters are
temporarily replaced when a shell function is executed (see
.SM
.B FUNCTIONS
below).
.PP
When a positional parameter consisting of more than a single
digit is expanded, it must be enclosed in braces (see
.SM
.B EXPANSION
below).
.SS Special Parameters
.PP
The shell treats several parameters specially. These parameters may
only be referenced; assignment to them is not allowed.
.PD 0
.TP
.B *
Expands to the positional parameters, starting from one. When the
expansion occurs within double quotes, it expands to a single word
with the value of each parameter separated by the first character
of the
.SM
.B IFS
special variable. That is, "\fB$*\fP" is equivalent
to "\fB$1\fP\fIc\fP\fB$2\fP\fIc\fP\fB...\fP", where
.I c
is the first character of the value of the
.SM
.B IFS
variable. If
.SM
.B IFS
is unset, the parameters are separated by spaces.
If

```
.SM
.B IFS
is null, the parameters are joined without intervening separators.
.TP
.B @
Expands to the positional parameters, starting from one.  When the
expansion occurs within double quotes, each parameter expands to a
separate word.  That is, "\fB$@\fP" is equivalent to
"\fB$1\fP" "\fB$2\fP" ...
If the double-quoted expansion occurs within a word, the expansion of
the first parameter is joined with the beginning part of the original
word, and the expansion of the last parameter is joined with the last
part of the original word.
When there are no positional parameters, "\fB$@\fP" and
.B $@
expand to nothing (i.e., they are removed).
.TP
.B #
Expands to the number of positional parameters in decimal.
.TP
.B ?
Expands to the status of the most recently executed foreground
pipeline.
.TP
.B \-
Expands to the current option flags as specified upon invocation,
by the
.B set
builtin command, or those set by the shell itself
(such as the
.B \-i
option).
.TP
.B $
Expands to the process ID of the shell.  In a () subshell, it
expands to the process ID of the current shell, not the
subshell.
.TP
.B !
Expands to the process ID of the most recently executed background
(asynchronous) command.
.TP
.B 0
Expands to the name of the shell or shell script.  This is set at
shell initialization.  If
.B bash
is invoked with a file of commands,
.B $0
is set to the name of that file.  If
.B bash
is started with the
.B \-c
option, then
.B $0
is set to the first argument after the string to be
executed, if one is present.  Otherwise, it is set
to the file name used to invoke
.BR bash ,
as given by argument zero.
.TP
.B _
At shell startup, set to the absolute pathname used to invoke the
shell or shell script being executed as passed in the environment
or argument list.
Subsequently, expands to the last argument to the previous command,
after expansion.
Also set to the full pathname used to invoke each command executed
and placed in the environment exported to that command.
When checking mail, this parameter holds the name of the mail file
currently being checked.
.PD
.SS Shell Variables
.PP
The following variables are set by the shell:
.PP
.PD 0
.TP
.B BASH
Expands to the full file name used to invoke this instance of
.BR bash .
.TP
.B BASH_ARGC
An array variable whose values are the number of parameters in each
frame of the current bash execution call stack.
The number of
parameters to the current subroutine (shell function or script executed
with \fB.\fP or \fBsource\fP) is at the top of the stack.
When a subroutine is executed, the number of parameters passed is pushed onto
\fBBASH_ARGC\fP.
The shell sets \fBBASH_ARGC\fP only when in extended debugging mode
(see the description of the
.B extdebug
option to the
.B shopt
builtin below)
.TP
.B BASH_ARGV
An array variable containing all of the parameters in the current bash
execution call stack.  The final parameter of the last subroutine call
is at the top of the stack; the first parameter of the initial call is
at the bottom.  When a subroutine is executed, the parameters supplied
are pushed onto \fBBASH_ARGV\fP.
The shell sets \fBBASH_ARGV\fP only when in extended debugging mode
(see the description of the
.B extdebug
option to the
.B shopt
builtin below)
.TP
.B BASH_COMMAND
The command currently being executed or about to be executed, unless the
shell is executing a command as the result of a trap,
in which case it is the command executing at the time of the trap.
.TP
.B BASH_EXECUTION_STRING
The command argument to the \fB\-c\fP invocation option.
.TP
.B BASH_LINENO
An array variable whose members are the line numbers in source files
corresponding to each member of \fBFUNCNAME\fP.
\fB${BASH_LINENO[\fP\fI$i\fP\fB]}\fP is the line number in the source
file where \fB${FUNCNAME[\fP\fI$i\fP\fB]}\fP was called.
```

The corresponding source file name is \fB${BASH_SOURCE[\fP\fI$i\fP\fB]}\fB.
Use \fBLINENO\fP to obtain the current line number.
.TP
.B BASH_REMATCH
An array variable whose members are assigned by the \fB=~\fP binary
operator to the \fB[[\fP conditional command.
The element with index 0 is the portion of the string
matching the entire regular expression.
The element with index \fIn\fP is the portion of the
string matching the \fIn\fPth parenthesized subexpression.
This variable is read-only.
.TP
.B BASH_SOURCE
An array variable whose members are the source filenames corresponding
to the elements in the \fBFUNCNAME\fP array variable.
.TP
.B BASH_SUBSHELL
Incremented by one each time a subshell or subshell environment is spawned.
The initial value is 0.
.TP
.B BASH_VERSINFO
A readonly array variable whose members hold version information for
this instance of
.BR bash .
The values assigned to the array members are as follows:
.sp .5
.RS
.PD 0
.TP 24
.B BASH_VERSINFO[\fR0\fP]
The major version number (the \fIrelease\fP).
.TP
.B BASH_VERSINFO[\fR1\fP]
The minor version number (the \fIversion\fP).
.TP
.B BASH_VERSINFO[\fR2\fP]
The patch level.
.TP
.B BASH_VERSINFO[\fR3\fP]
The build version.
.TP
.B BASH_VERSINFO[\fR4\fP]
The release status (e.g., \fIbeta1\fP).
.TP
.B BASH_VERSINFO[\fR5\fP]
The value of \fBMACHTYPE\fP.
.PD
.RE
.TP
.B BASH_VERSION
Expands to a string describing the version of this instance of
.BR bash .
.TP
.B COMP_CWORD
An index into \fB${COMP_WORDS}\fP of the word containing the current
cursor position.
This variable is available only in shell functions invoked by the
programmable completion facilities (see \fBProgrammable Completion\fP
below).
.TP
.B COMP_LINE
The current command line.
This variable is available only in shell functions and external
commands invoked by the
programmable completion facilities (see \fBProgrammable Completion\fP
below).
.TP
.B COMP_POINT
The index of the current cursor position relative to the beginning of
the current command.
If the current cursor position is at the end of the current command,
the value of this variable is equal to \fB${#COMP_LINE}\fP.
This variable is available only in shell functions and external
commands invoked by the
programmable completion facilities (see \fBProgrammable Completion\fP
below).
.TP
.B COMP_WORDBREAKS
The set of characters that the Readline library treats as word
separators when performing word completion.
If
.SM
.B COMP_WORDBREAKS
is unset, it loses its special properties, even if it is
subsequently reset.
.TP
.B COMP_WORDS
An array variable (see \fBArrays\fP below) consisting of the individual
words in the current command line.
The words are split on shell metacharacters as the shell parser would
separate them.
This variable is available only in shell functions invoked by the
programmable completion facilities (see \fBProgrammable Completion\fP
below).
.TP
.B DIRSTACK
An array variable (see
.B Arrays
below) containing the current contents of the directory stack.
Directories appear in the stack in the order they are displayed by the
.B dirs
builtin.
Assigning to members of this array variable may be used to modify
directories already in the stack, but the
.B pushd
and
.B popd
builtins must be used to add and remove directories.
Assignment to this variable will not change the current directory.
If
.SM
.B DIRSTACK
is unset, it loses its special properties, even if it is
subsequently reset.
.TP
.B EUID
Expands to the effective user ID of the current user, initialized at
shell startup. This variable is readonly.
.TP
.B FUNCNAME
An array variable containing the names of all shell functions
currently in the execution call stack.

The element with index 0 is the name of any currently-executing
shell function.
The bottom-most element is "main".
This variable exists only when a shell function is executing.
Assignments to
.SM
.B FUNCNAME
have no effect and return an error status.
If
.SM
.B FUNCNAME
is unset, it loses its special properties, even if it is
subsequently reset.
.TP
.B GROUPS
An array variable containing the list of groups of which the current
user is a member.
Assignments to
.SM
.B GROUPS
have no effect and return an error status.
If
.SM
.B GROUPS
is unset, it loses its special properties, even if it is
subsequently reset.
.TP
.B HISTCMD
The history number, or index in the history list, of the current
command.
If
.SM
.B HISTCMD
is unset, it loses its special properties, even if it is
subsequently reset.
.TP
.B HOSTNAME
Automatically set to the name of the current host.
.TP
.B HOSTTYPE
Automatically set to a string that uniquely
describes the type of machine on which
.B bash
is executing.
The default is system-dependent.
.TP
.B LINENO
Each time this parameter is referenced, the shell substitutes
a decimal number representing the current sequential line number
(starting with 1) within a script or function. When not in a
script or function, the value substituted is not guaranteed to
be meaningful.
If
.SM
.B LINENO
is unset, it loses its special properties, even if it is
subsequently reset.
.TP
.B MACHTYPE
Automatically set to a string that fully describes the system
type on which
.B bash
is executing, in the standard GNU \fIcpu-company-system\fP format.
The default is system-dependent.
.TP
.B OLDPWD
The previous working directory as set by the
.B cd
command.
.TP
.B OPTARG
The value of the last option argument processed by the
.B getopts
builtin command (see
.SM
.B SHELL BUILTIN COMMANDS
below).
.TP
.B OPTIND
The index of the next argument to be processed by the
.B getopts
builtin command (see
.SM
.B SHELL BUILTIN COMMANDS
below).
.TP
.B OSTYPE
Automatically set to a string that
describes the operating system on which
.B bash
is executing.
The default is system-dependent.
.TP
.B PIPESTATUS
An array variable (see
.B Arrays
below) containing a list of exit status values from the processes
in the most-recently-executed foreground pipeline (which may
contain only a single command).
.TP
.B PPID
The process ID of the shell's parent. This variable is readonly.
.TP
.B PWD
The current working directory as set by the
.B cd
command.
.TP
.B RANDOM
Each time this parameter is referenced, a random integer between
0 and 32767 is
generated. The sequence of random numbers may be initialized by assigning
a value to
.SM
.BR RANDOM .
If
.SM
.B RANDOM
is unset, it loses its special properties, even if it is
subsequently reset.
.TP
.B REPLY

Set to the line of input read by the
.B read
builtin command when no arguments are supplied.
.TP
.B SECONDS
Each time this parameter is
referenced, the number of seconds since shell invocation is returned. If a
value is assigned to
.SM
.BR SECONDS ,
the value returned upon subsequent
references is
the number of seconds since the assignment plus the value assigned.
If
.SM
.B SECONDS
is unset, it loses its special properties, even if it is
subsequently reset.
.TP
.B SHELLOPTS
A colon-separated list of enabled shell options. Each word in
the list is a valid argument for the
.B \-o
option to the
.B set
builtin command (see
.SM
.B "SHELL BUILTIN COMMANDS"
below). The options appearing in
.SM
.B SHELLOPTS
are those reported as
.I on
by \fBset \-o\fP.
If this variable is in the environment when
.B bash
starts up, each shell option in the list will be enabled before
reading any startup files.
This variable is read-only.
.TP
.B SHLVL
Incremented by one each time an instance of
.B bash
is started.
.TP
.B UID
Expands to the user ID of the current user, initialized at shell startup.
This variable is readonly.
.PD
.PP
The following variables are used by the shell. In some cases,
.B bash
assigns a default value to a variable; these cases are noted
below.
.PP
.PD 0
.TP
.B BASH_ENV
If this parameter is set when \fBbash\fP is executing a shell script,
its value is interpreted as a filename containing commands to
initialize the shell, as in
.IR ~/.bashrc .
The value of
.SM
.B BASH_ENV
is subjected to parameter expansion, command substitution, and arithmetic
expansion before being interpreted as a file name.
.SM
.B PATH
is not used to search for the resultant file name.
.TP
.B CDPATH
The search path for the
.B cd
command.
This is a colon-separated list of directories in which the shell looks
for destination directories specified by the
.B cd
command.
A sample value is
.if t \f(CW".:~:/usr"\fP.
.if n ".:~:/usr".
.TP
.B COLUMNS
Used by the \fBselect\fP builtin command to determine the terminal width
when printing selection lists. Automatically set upon receipt of a SIGWINCH.
.TP
.B COMPREPLY
An array variable from which \fBbash\fP reads the possible completions
generated by a shell function invoked by the programmable completion
facility (see \fBProgrammable Completion\fP below).
.TP
.B EMACS
If \fBbash\fP finds this variable in the environment when the shell starts
with value
.if t \f(CWt\fP,
.if n "t",
it assumes that the shell is running in an emacs shell buffer and disables
line editing.
.TP
.B FCEDIT
The default editor for the
.B fc
builtin command.
.TP
.B FIGNORE
A colon-separated list of suffixes to ignore when performing
filename completion (see
.SM
.B READLINE
below).
A filename whose suffix matches one of the entries in
.SM
.B FIGNORE
is excluded from the list of matched filenames.
A sample value is
.if t \f(CW".o:~"\fP.
.if n ".o:~".
.TP
.B GLOBIGNORE
A colon-separated list of patterns defining the set of filenames to
be ignored by pathname expansion.

If a filename matched by a pathname expansion pattern also matches one
of the patterns in
.SM
.BR GLOBIGNORE ,
it is removed from the list of matches.
.TP
.B HISTCONTROL
A colon-separated list of values controlling how commands are saved on
the history list.
If the list of values includes
.IR ignorespace ,
lines which begin with a
.B space
character are not saved in the history list.
A value of
.I ignoredups
causes lines matching the previous history entry to not be saved.
A value of
.I ignoreboth
is shorthand for \fIignorespace\fP and \fIignoredups\fP.
A value of
.IR erasedups
causes all previous lines matching the current line to be removed from
the history list before that line is saved.
Any value not in the above list is ignored.
If \fBHISTCONTROL\fP is unset, or does not include a valid value,
all lines read by the shell parser are saved on the history list,
subject to the value of
.BR HISTIGNORE .
The second and subsequent lines of a multi-line compound command are
not tested, and are added to the history regardless of the value of
.BR HISTCONTROL .
.TP
.B HISTFILE
The name of the file in which command history is saved (see
.SM
.B HISTORY
below). The default value is \fI~/.bash_history\fP. If unset, the
command history is not saved when an interactive shell exits.
.TP
.B HISTFILESIZE
The maximum number of lines contained in the history file. When this
variable is assigned a value, the history file is truncated, if
necessary, by removing the oldest entries,
to contain no more than that number of lines. The default
value is 500. The history file is also truncated to this size after
writing it when an interactive shell exits.
.TP
.B HISTIGNORE
A colon-separated list of patterns used to decide which command lines
should be saved on the history list. Each pattern is anchored at the
beginning of the line and must match the complete line (no implicit
`\fB*\fP' is appended). Each pattern is tested against the line
after the checks specified by
.B HISTCONTROL
are applied.
In addition to the normal shell pattern matching characters, `\fB&\fP'
matches the previous history line. `\fB&\fP' may be escaped using a
backslash; the backslash is removed before attempting a match.
The second and subsequent lines of a multi-line compound command are
not tested, and are added to the history regardless of the value of
.BR HISTIGNORE .
.TP
.B HISTSIZE
The number of commands to remember in the command history (see
.SM
.B HISTORY
below). The default value is 500.
.TP
.B HISTTIMEFORMAT
If this variable is set and not null, its value is used as a format string
for \fIstrftime\fP(3) to print the time stamp associated with each history
entry displayed by the \fBhistory\fP builtin.
If this variable is set, time stamps are written to the history file so
they may be preserved across shell sessions.
.TP
.B HOME
The home directory of the current user; the default argument for the
\fBcd\fP builtin command.
The value of this variable is also used when performing tilde expansion.
.TP
.B HOSTFILE
Contains the name of a file in the same format as
.FN /etc/hosts
that should be read when the shell needs to complete a
hostname.
The list of possible hostname completions may be changed while the
shell is running;
the next time hostname completion is attempted after the
value is changed,
.B bash
adds the contents of the new file to the existing list.
If
.SM
.B HOSTFILE
is set, but has no value, \fBbash\fP attempts to read
.FN /etc/hosts
to obtain the list of possible hostname completions.
When
.SM
.B HOSTFILE
is unset, the hostname list is cleared.
.TP
.B IFS
The
.I Internal Field Separator
that is used
for word splitting after expansion and to
split lines into words with the
.B read
builtin command. The default value is
``<space><tab><newline>''.
.TP
.B IGNOREEOF
Controls the
action of an interactive shell on receipt of an
.SM
.B EOF
character as the sole input. If set, the value is the number of
consecutive
.SM
.B EOF

characters which must be
typed as the first characters on an input line before
.B bash
exits. If the variable exists but does not have a numeric value, or
has no value, the default value is 10. If it does not exist,
.SM
.B EOF
signifies the end of input to the shell.
.TP
.B INPUTRC
The filename for the
.B readline
startup file, overriding the default of
.FN ~/.inputrc
(see
.SM
.B READLINE
below).
.TP
.B LANG
Used to determine the locale category for any category not specifically
selected with a variable starting with \fBLC_\fP.
.TP
.B LC_ALL
This variable overrides the value of \fBLANG\fP and any other
\fBLC_\fP variable specifying a locale category.
.TP
.B LC_COLLATE
This variable determines the collation order used when sorting the
results of pathname expansion, and determines the behavior of range
expressions, equivalence classes, and collating sequences within
pathname expansion and pattern matching.
.TP
.B LC_CTYPE
This variable determines the interpretation of characters and the
behavior of character classes within pathname expansion and pattern
matching.
.TP
.B LC_MESSAGES
This variable determines the locale used to translate double-quoted
strings preceded by a \fB$\fP.
.TP
.B LC_NUMERIC
This variable determines the locale category used for number formatting.
.TP
.B LINES
Used by the \fBselect\fP builtin command to determine the column length
for printing selection lists. Automatically set upon receipt of a SIGWINCH.
.TP
.B MAIL
If this parameter is set to a file name and the
.SM
.B MAILPATH
variable is not set,
.B bash
informs the user of the arrival of mail in the specified file.
.TP
.B MAILCHECK
Specifies how
often (in seconds)
.B bash
checks for mail. The default is 60 seconds. When it is time to check
for mail, the shell does so before displaying the primary prompt.
If this variable is unset, or set to a value that is not a number
greater than or equal to zero, the shell disables mail checking.
.TP
.B MAILPATH
A colon-separated list of file names to be checked for mail.
The message to be printed when mail arrives in a particular file
may be specified by separating the file name from the message with a `?'.
When used in the text of the message, \fB$_\fP expands to the name of
the current mailfile.
Example:
.RS
.PP
\fBMAILPATH\fP=\(aq/var/mail/bfox?"You have mail":~/shell\-mail?"$_ has mail!"\(aq
.PP
.B Bash
supplies a default value for this variable, but the location of the user
mail files that it uses is system dependent (e.g., /var/mail/\fB$USER\fP).
.RE
.TP
.B OPTERR
If set to the value 1,
.B bash
displays error messages generated by the
.B getopts
builtin command (see
.SM
.B SHELL BUILTIN COMMANDS
below).
.SM
.B OPTERR
is initialized to 1 each time the shell is invoked or a shell
script is executed.
.TP
.B PATH
The search path for commands. It
is a colon-separated list of directories in which
the shell looks for commands (see
.SM
.B COMMAND EXECUTION
below).
A zero-length (null) directory name in the value of \fBPATH\fP indicates the
current directory.
A null directory name may appear as two adjacent colons, or as an initial
or trailing colon.
The default path is system-dependent,
and is set by the administrator who installs
.BR bash .
A common value is
.if t \f(CW/usr/gnu/bin:/usr/local/bin:/usr/ucb:/bin:/usr/bin\fP.
.if n ``/usr/gnu/bin:/usr/local/bin:/usr/ucb:/bin:/usr/bin''.
.TP
.B POSIXLY_CORRECT
If this variable is in the environment when \fBbash\fP starts, the shell
enters \fIposix mode\fP before reading the startup files, as if the
.B \-\-posix
invocation option had been supplied. If it is set while the shell is
running, \fBbash\fP enables \fIposix mode\fP, as if the command
.if t \f(CWset -o posix\fP
.if n \fIset -o posix\fP

had been executed.
.TP
.B PROMPT_COMMAND
If set, the value is executed as a command prior to issuing each primary
prompt.
.TP
.B PS1
The value of this parameter is expanded (see
.SM
.B PROMPTING
below) and used as the primary prompt string. The default value is
``\fB\es\-\ev\e$ \fP''.
.TP
.B PS2
The value of this parameter is expanded as with
.B PS1
and used as the secondary prompt string. The default is
``\fB> \fP''.
.TP
.B PS3
The value of this parameter is used as the prompt for the
.B select
command (see
.SM
.B SHELL GRAMMAR
above).
.TP
.B PS4
The value of this parameter is expanded as with
.B PS1
and the value is printed before each command
.B bash
displays during an execution trace. The first character of
.SM
.B PS4
is replicated multiple times, as necessary, to indicate multiple
levels of indirection. The default is ``\fB+ \fP''.
.TP
.B SHELL
The full pathname to the shell is kept in this environment variable.
If it is not set when the shell starts,
.B bash
assigns to it the full pathname of the current user's login shell.
.TP
.B TIMEFORMAT
The value of this parameter is used as a format string specifying
how the timing information for pipelines prefixed with the
.B time
reserved word should be displayed.
The \fB%\fP character introduces an escape sequence that is
expanded to a time value or other information.
The escape sequences and their meanings are as follows; the
braces denote optional portions.
.sp .5
.RS
.PD 0
.TP 10
.B %%
A literal \fB%\fP.
.TP
.B %[\fIp\fP][l]R
The elapsed time in seconds.
.TP
.B %[\fIp\fP][l]U
The number of CPU seconds spent in user mode.
.TP
.B %[\fIp\fP][l]S
The number of CPU seconds spent in system mode.
.TP
.B %P
The CPU percentage, computed as (%U + %S) / %R.
.PD
.RE
.IP
The optional \fIp\fP is a digit specifying the \fIprecision\fP,
the number of fractional digits after a decimal point.
A value of 0 causes no decimal point or fraction to be output.
At most three places after the decimal point may be specified;
values of \fIp\fP greater than 3 are changed to 3.
If \fIp\fP is not specified, the value 3 is used.
.IP
The optional \fBl\fP specifies a longer format, including
minutes, of the form \fIMM\fPm\fISS\fP.\fIFF\fPs.
The value of \fIp\fP determines whether or not the fraction is
included.
.IP
If this variable is not set, \fBbash\fP acts as if it had the
value \fB$\(aq\enreal\et%3lR\enuser\et%3lU\ensys\t%3lS\(aq\fP.
If the value is null, no timing information is displayed.
A trailing newline is added when the format string is displayed.
.TP
.B TMOUT
If set to a value greater than zero, \fBTMOUT\fP is treated as the
default timeout for the \fBread\fP builtin.
The \fBselect\fP command terminates if input does not arrive
after \fBTMOUT\fP seconds when input is coming from a terminal.
In an interactive shell, the value is interpreted as the
number of seconds to wait for input after issuing the primary prompt.
.B Bash
terminates after waiting for that number of seconds if input does
not arrive.
.TP
.B TMPDIR
If set, \fBBash\fP uses its value as the name of a directory in which
\fBBash\fP creates temporary files for the shell's use.
.TP
.B auto_resume
This variable controls how the shell interacts with the user and
job control. If this variable is set, single word simple
commands without redirections are treated as candidates for resumption
of an existing stopped job. There is no ambiguity allowed; if there is
more than one job beginning with the string typed, the job most recently
accessed is selected. The
.I name
of a stopped job, in this context, is the command line used to
start it.
If set to the value
.IR exact ,
the string supplied must match the name of a stopped job exactly;
if set to
.IR substring ,
the string supplied needs to match a substring of the name of a

stopped job. The
.I substring
value provides functionality analogous to the
.B %?
job identifier (see
.SM
.B JOB CONTROL
below). If set to any other value, the supplied string must
be a prefix of a stopped job's name; this provides functionality
analogous to the \fB%\fP\fIstring\fP job identifier.
.TP
.B histchars
The two or three characters which control history expansion
and tokenization (see
.SM
.B HISTORY EXPANSION
below). The first character is the \fIhistory expansion\fP character,
the character which signals the start of a history
expansion, normally `\fB!\fP'.
The second character is the \fIquick substitution\fP
character, which is used as shorthand for re-running the previous
command entered, substituting one string for another in the command.
The default is `\fB^\fP'.
The optional third character is the character
which indicates that the remainder of the line is a comment when found
as the first character of a word, normally `\fB#\fP'. The history
comment character causes history substitution to be skipped for the
remaining words on the line. It does not necessarily cause the shell
parser to treat the rest of the line as a comment.
.PD
.SS Arrays
.B Bash
provides one-dimensional array variables. Any variable may be used as
an array; the
.B declare
builtin will explicitly declare an array. There is no maximum
limit on the size of an array, nor any requirement that members
be indexed or assigned contiguously. Arrays are indexed using
integers and are zero-based.
.PP
An array is created automatically if any variable is assigned to using
the syntax \fIname\fP[\fIsubscript\fP]=\fIvalue\fP. The
.I subscript
is treated as an arithmetic expression that must evaluate to a number
greater than or equal to zero. To explicitly declare an array, use
.B declare \-a \fIname\fP
(see
.SM
.B SHELL BUILTIN COMMANDS
below).
.B declare \-a \fIname\fP[\fIsubscript\fP]
is also accepted; the \fIsubscript\fP is ignored. Attributes may be
specified for an array variable using the
.B declare
and
.B readonly
builtins. Each attribute applies to all members of an array.
.PP
Arrays are assigned to using compound assignments of the form
\fIname\fP=\fB(\fPvalue\fI1\fP ... value\fIn\fP\fB)\fP, where each
\fIvalue\fP is of the form [\fIsubscript\fP]=\fIstring\fP. Only
\fIstring\fP is required. If
the optional brackets and subscript are supplied, that index is assigned to;
otherwise the index of the element assigned is the last index assigned
to by the statement plus one. Indexing starts at zero.
This syntax is also accepted by the
.B declare
builtin. Individual array elements may be assigned to using the
\fIname\fP[\fIsubscript\fP]=\fIvalue\fP syntax introduced above.
.PP
Any element of an array may be referenced using
${\fIname\fP[\fIsubscript\fP]}. The braces are required to avoid
conflicts with pathname expansion. If
\fIsubscript\fP is \fB@\fP or \fB*\fP, the word expands to
all members of \fIname\fP. These subscripts differ only when the
word appears within double quotes. If the word is double-quoted,
${\fIname\fP[*]} expands to a single
word with the value of each array member separated by the first
character of the
.SM
.B IFS
special variable, and ${\fIname\fP[@]} expands each element of
\fIname\fP to a separate word. When there are no array members,
${\fIname\fP[@]} expands to nothing.
If the double-quoted expansion occurs within a word, the expansion of
the first parameter is joined with the beginning part of the original
word, and the expansion of the last parameter is joined with the last
part of the original word.
This is analogous to the expansion
of the special parameters \fB*\fP and \fB@\fP (see
.B Special Parameters
above). ${#\fIname\fP[\fIsubscript\fP]} expands to the length of
${\fIname\fP[\fIsubscript\fP]}. If \fIsubscript\fP is \fB*\fP or
\fB@\fP, the expansion is the number of elements in the array.
Referencing an array variable without a subscript is equivalent to
referencing element zero.
.PP
The
.B unset
builtin is used to destroy arrays. \fBunset\fP \fIname\fP[\fIsubscript\fP]
destroys the array element at index \fIsubscript\fP.
Care must be taken to avoid unwanted side effects caused by filename
generation.
\fBunset\fP \fIname\fP, where \fIname\fP is an array, or
\fBunset\fP \fIname\fP[\fIsubscript\fP], where
\fIsubscript\fP is \fB*\fP or \fB@\fP, removes the entire array.
.PP
The
.BR declare ,
.BR local ,
and
.B readonly
builtins each accept a
.B \-a
option to specify an array. The
.B read
builtin accepts a
.B \-a
option to assign a list of words read from the standard input
to an array. The
.B set
and

.B declare
builtins display array values in a way that allows them to be
reused as assignments.
.SH EXPANSION
Expansion is performed on the command line after it has been split into
words. There are seven kinds of expansion performed:
.IR "brace expansion" ,
.IR "tilde expansion" ,
.IR "parameter and variable expansion" ,
.IR "command substitution" ,
.IR "arithmetic expansion" ,
.IR "word splitting" ,
and
.IR "pathname expansion" .
.PP
The order of expansions is: brace expansion, tilde expansion,
parameter, variable and arithmetic expansion and
command substitution
(done in a left-to-right fashion), word splitting, and pathname
expansion.
.PP
On systems that can support it, there is an additional expansion
available: \fIprocess substitution\fP.
.PP
Only brace expansion, word splitting, and pathname expansion
can change the number of words of the expansion; other expansions
expand a single word to a single word.
The only exceptions to this are the expansions of
"\fB$@\fP" and "\fB${\fP\fIname\fP\fB[@]}\fP"
as explained above (see
.SM
.BR PARAMETERS).
.SS Brace Expansion
.PP
.I "Brace expansion"
is a mechanism by which arbitrary strings
may be generated. This mechanism is similar to
\fIpathname expansion\fP, but the filenames generated
need not exist. Patterns to be brace expanded take
the form of an optional
.IR preamble ,
followed by either a series of comma-separated strings or
a sequence expression between a pair of braces, followed by
an optional
.IR postscript .
The preamble is prefixed to each string contained
within the braces, and the postscript is then appended
to each resulting string, expanding left to right.
.PP
Brace expansions may be nested. The results of each expanded
string are not sorted; left to right order is preserved.
For example, a\fB{\fPd,c,b\fB}\fPe expands into `ade ace abe'.
.PP
A sequence expression takes the form \fB{\fP\fIx\fP\fB..\fP\fIy\fP\fB}\fP,
where \fIx\fP and \fIy\fP are either integers or single characters.
When integers are supplied, the expression expands to each number between
\fIx\fP and \fIy\fP, inclusive.
When characters are supplied, the expression expands to each character
lexicographically between \fIx\fP and \fIy\fP, inclusive. Note that
both \fIx\fP and \fIy\fP must be of the same type.
.PP
Brace expansion is performed before any other expansions,
and any characters special to other expansions are preserved
in the result. It is strictly textual.
.B Bash
does not apply any syntactic interpretation to the context of the
expansion or the text between the braces.
.PP
A correctly-formed brace expansion must contain unquoted opening
and closing braces, and at least one unquoted comma or a valid
sequence expression.
Any incorrectly formed brace expansion is left unchanged.
A \fB{\fP or \fB,\fP may be quoted with a backslash to prevent its
being considered part of a brace expression.
To avoid conflicts with parameter expansion, the string \fB${\fP
is not considered eligible for brace expansion.
.PP
This construct is typically used as shorthand when the common
prefix of the strings to be generated is longer than in the
above example:
.RS
.PP
mkdir /usr/local/src/bash/{old,new,dist,bugs}
.RE
or
.RS
chown root /usr/{ucb/{ex,edit},lib/{ex?.?*,how_ex}}
.RE
.PP
Brace expansion introduces a slight incompatibility with
historical versions of
.BR sh .
.B sh
does not treat opening or closing braces specially when they
appear as part of a word, and preserves them in the output.
.B Bash
removes braces from words as a consequence of brace
expansion. For example, a word entered to
.B sh
as \fIfile{1,2}\fP
appears identically in the output. The same word is
output as
.I file1 file2
after expansion by
.BR bash .
If strict compatibility with
.B sh
is desired, start
.B bash
with the
.B +B
option or disable brace expansion with the
.B +B
option to the
.B set
command (see
.SM
.B SHELL BUILTIN COMMANDS
below).
.SS Tilde Expansion
.PP
If a word begins with an unquoted tilde character (`\fB~\fP'), all of

the characters preceding the first unquoted slash (or all characters,
if there is no unquoted slash) are considered a \fItilde-prefix\fP.
If none of the characters in the tilde-prefix are quoted, the
characters in the tilde-prefix following the tilde are treated as a
possible \fIlogin name\fP.
If this login name is the null string, the tilde is replaced with the
value of the shell parameter
.SM
.BR HOME .
If
.SM
.B HOME
is unset, the home directory of the user executing the shell is
substituted instead.
Otherwise, the tilde-prefix is replaced with the home directory
associated with the specified login name.
.PP
If the tilde-prefix is a `~+', the value of the shell variable
.SM
.B PWD
replaces the tilde-prefix.
If the tilde-prefix is a `~\-', the value of the shell variable
.SM
.BR OLDPWD ,
if it is set, is substituted.
If the characters following the tilde in the tilde-prefix consist
of a number \fIN\fP, optionally prefixed
by a `+' or a `\-', the tilde-prefix is replaced with the corresponding
element from the directory stack, as it would be displayed by the
.B dirs
builtin invoked with the tilde-prefix as an argument.
If the characters following the tilde in the tilde-prefix consist of a
number without a leading `+' or `\-', `+' is assumed.
.PP
If the login name is invalid, or the tilde expansion fails, the word
is unchanged.
.PP
Each variable assignment is checked for unquoted tilde-prefixes immediately
following a
.B :
or the first
.BR = .
In these cases, tilde expansion is also performed.
Consequently, one may use file names with tildes in assignments to
.SM
.BR PATH ,
.SM
.BR MAILPATH ,
and
.SM
.BR CDPATH ,
and the shell assigns the expanded value.
.SS Parameter Expansion
.PP
The `\fB$\fP' character introduces parameter expansion,
command substitution, or arithmetic expansion. The parameter name
or symbol to be expanded may be enclosed in braces, which
are optional but serve to protect the variable to be expanded from
characters immediately following it which could be
interpreted as part of the name.
.PP
When braces are used, the matching ending brace is the first `\fB}\fP'
not escaped by a backslash or within a quoted string, and not within an
embedded arithmetic expansion, command substitution, or parameter
expansion.
.PP
.PD 0
.TP
${\fIparameter\fP}
The value of \fIparameter\fP is substituted. The braces are required
when
.I parameter
is a positional parameter with more than one digit,
or when
.I parameter
is followed by a character which is not to be
interpreted as part of its name.
.PD
.PP
If the first character of \fIparameter\fP is an exclamation point,
a level of variable indirection is introduced.
\fBBash\fP uses the value of the variable formed from the rest of
\fIparameter\fP as the name of the variable; this variable is then
expanded and that value is used in the rest of the substitution, rather
than the value of \fIparameter\fP itself.
This is known as \fIindirect expansion\fP.
The exceptions to this are the expansions of ${!\fIprefix\fP*} and
${\fB!\fP\fIname\fP[\fI@\fP]} described below.
The exclamation point must immediately follow the left brace in order to
introduce indirection.
.PP
In each of the cases below, \fIword\fP is subject to tilde expansion,
parameter expansion, command substitution, and arithmetic expansion.
When not performing substring expansion, \fBbash\fP tests for a parameter
that is unset or null; omitting the colon results in a test only for a
parameter that is unset.
.PP
.PD 0
.TP
${\fIparameter\fP\fB:\-\fP\fIword\fP}
\fBUse Default Values\fP. If
.I parameter
is unset or null, the expansion of
.I word
is substituted. Otherwise, the value of
.I parameter
is substituted.
.TP
${\fIparameter\fP\fB:=\fP\fIword\fP}
\fBAssign Default Values\fP.
If
.I parameter
is unset or null, the expansion of
.I word
is assigned to
.IR parameter .
The value of
.I parameter
is then substituted. Positional parameters and special parameters may
not be assigned to in this way.
.TP
${\fIparameter\fP\fB:?\fP\fIword\fP}

\fBDisplay Error if Null or Unset\fP.
If
.I parameter
is null or unset, the expansion of \fIword\fP (or a message to that effect
if
.I word
is not present) is written to the standard error and the shell, if it
is not interactive, exits. Otherwise, the value of \fIparameter\fP is
substituted.
.TP
${\fIparameter\fP\fB:+\fP\fIword\fP}
\fBUse Alternate Value\fP.
If
.I parameter
is null or unset, nothing is substituted, otherwise the expansion of
.I word
is substituted.
.TP
${\fIparameter\fP\fB:\fP\fIoffset\fP}
.PD 0
.TP
${\fIparameter\fP\fB:\fP\fIoffset\fP\fB:\fP\fIlength\fP}
.PD
\fBSubstring Expansion.\fP
Expands to up to \fIlength\fP characters of \fIparameter\fP
starting at the character specified by \fIoffset\fP.
If \fIlength\fP is omitted, expands to the substring of
\fIparameter\fP starting at the character specified by \fIoffset\fP.
\fIlength\fP and \fIoffset\fP are arithmetic expressions (see
.SM
.B
ARITHMETIC EVALUATION
below).
\fIlength\fP must evaluate to a number greater than or equal to zero.
If \fIoffset\fP evaluates to a number less than zero, the value
is used as an offset from the end of the value of \fIparameter\fP.
If \fIparameter\fP is \fB@\fP, the result is \fIlength\fP positional
parameters beginning at \fIoffset\fP.
If \fIparameter\fP is an array name indexed by @ or *,
the result is the \fIlength\fP
members of the array beginning with ${\fIparameter\fP[\fIoffset\fP]}.
A negative \fIoffset\fP is taken relative to one greater than the maximum
index of the specified array.
Note that a negative offset must be separated from the colon by at least
one space to avoid being confused with the :- expansion.
Substring indexing is zero-based unless the positional parameters
are used, in which case the indexing starts at 1.
.TP
${\fB!\fP\fIprefix\fP\fB*\fP}
.PD 0
.TP
${\fB!\fP\fIprefix\fP\fB@\fP}
.PD
Expands to the names of variables whose names begin with \fIprefix\fP,
separated by the first character of the
.SM
.B IFS
special variable.
.TP
${\fB!\fP\fIname\fP[\fI@\fP]}
.PD 0
.TP
${\fB!\fP\fIname\fP[\fI*\fP]}
.PD
If \fIname\fP is an array variable, expands to the list of array indices
(keys) assigned in \fIname\fP.
If \fIname\fP is not an array, expands to 0 if \fIname\fP is set and null
otherwise.
When \fI@\fP is used and the expansion appears within double quotes, each
key expands to a separate word.
.TP
${\fB#\fP\fIparameter\fP}
The length in characters of the value of \fIparameter\fP is substituted.
If
.I parameter
is
.B *
or
.BR @ ,
the value substituted is the number of positional parameters.
If
.I parameter
is an array name subscripted by
.B *
or
.BR @ ,
the value substituted is the number of elements in the array.
.TP
${\fIparameter\fP\fB#\fP\fIword\fP}
.PD 0
.TP
${\fIparameter\fP\fB##\fP\fIword\fP}
.PD
The
.I word
is expanded to produce a pattern just as in pathname
expansion. If the pattern matches the beginning of
the value of
.IR parameter ,
then the result of the expansion is the expanded value of
.I parameter
with the shortest matching pattern (the ``\fB#\fP'' case) or the
longest matching pattern (the ``\fB##\fP'' case) deleted.
If
.I parameter
is
.B @
or
.BR * ,
the pattern removal operation is applied to each positional
parameter in turn, and the expansion is the resultant list.
If
.I parameter
is an array variable subscripted with
.B @
or
.BR * ,
the pattern removal operation is applied to each member of the
array in turn, and the expansion is the resultant list.
.TP
${\fIparameter\fP\fB%\fP\fIword\fP}
.PD 0

```
.TP
${\fIparameter\fP\fB%%\fP\fIword\fP}
.PD
The \fIword\fP is expanded to produce a pattern just as in
pathname expansion.
If the pattern matches a trailing portion of the expanded value of
.IR parameter ,
then the result of the expansion is the expanded value of
.I parameter
with the shortest matching pattern (the ``\fB%\fP'' case) or the
longest matching pattern (the ``\fB%%\fP'' case) deleted.
If
.I parameter
is
.B @
or
.BR * ,
the pattern removal operation is applied to each positional
parameter in turn, and the expansion is the resultant list.
If
.I parameter
is an array variable subscripted with
.B @
or
.BR * ,
the pattern removal operation is applied to each member of the
array in turn, and the expansion is the resultant list.
.TP
${\fIparameter\fP\fB/\fP\fIpattern\fP\fB/\fP\fIstring\fP}
The \fIpattern\fP is expanded to produce a pattern just as in
pathname expansion.
\fIParameter\fP is expanded and the longest match of \fIpattern\fP
against its value is replaced with \fIstring\fP.
If \Ipattern\fP begins with \fB/\fP, all matches of \fIpattern\fP are
replaced with \fIstring\fP.  Normally only the first match is replaced.
If \fIpattern\fP begins with \fB#\fP, it must match at the beginning
of the expanded value of \fIparameter\fP.
If \fIpattern\fP begins with \fB%\fP, it must match at the end
of the expanded value of \fIparameter\fP.
If \fIstring\fP is null, matches of \fIpattern\fP are deleted
and the \fB/\fP following \fIpattern\fP may be omitted.
If
.I parameter
is
.B @
or
.BR * ,
the substitution operation is applied to each positional
parameter in turn, and the expansion is the resultant list.
If
.I parameter
is an array variable subscripted with
.B @
or
.BR * ,
the substitution operation is applied to each member of the
array in turn, and the expansion is the resultant list.
.SS Command Substitution
.PP
\fICommand substitution\fP allows the output of a command to replace
the command name.  There are two forms:
.PP
.RS
.PP
\fB$(\fP\fIcommand\fP\|\fB)\fP
.RE
or
.RS
\fB`\fP\fIcommand\fP\fB`\fP
.RE
.PP
.B Bash
performs the expansion by executing \fIcommand\fP and
replacing the command substitution with the standard output of the
command, with any trailing newlines deleted.
Embedded newlines are not deleted, but they may be removed during
word splitting.
The command substitution \fB$(cat \fIfile\fP)\fR can be replaced by
the equivalent but faster \fB$(< \fIfile\fP)\fR.
.PP
When the old-style backquote form of substitution is used,
backslash retains its literal meaning except when followed by
.BR $ ,
.BR ` ,
or
.BR \e .
The first backquote not preceded by a backslash terminates the
command substitution.
When using the $(\^\fIcommand\fP\|) form, all characters between the
parentheses make up the command; none are treated specially.
.PP
Command substitutions may be nested.  To nest when using the backquoted form,
escape the inner backquotes with backslashes.
.PP
If the substitution appears within double quotes, word splitting and
pathname expansion are not performed on the results.
.SS Arithmetic Expansion
.PP
Arithmetic expansion allows the evaluation of an arithmetic expression
and the substitution of the result.  The format for arithmetic expansion is:
.RS
.PP
\fB$((\fP\fIexpression\fP\fB))\fP
.RE
.PP
The
.I expression
is treated as if it were within double quotes, but a double quote
inside the parentheses is not treated specially.
All tokens in the expression undergo parameter expansion, string
expansion, command substitution, and quote removal.
Arithmetic expansions may be nested.
.PP
The evaluation is performed according to the rules listed below under
.SM
.BR "ARITHMETIC EVALUATION" .
If
.I expression
is invalid,
.B bash
prints a message indicating failure and no substitution occurs.
.SS Process Substitution
```

.PP
\fIProcess substitution\fP is supported on systems that support named
pipes (\fIFIFOs\fP) or the \fB/dev/fd\fP method of naming open files.
It takes the form of
\fB<(\fP\fIlist\^\fP\fB)\fP
or
\fB>(\fP\fIlist\^\fP\fB)\fP.
The process \fIlist\fP is run with its input or output connected to a
\fIFIFO\fP or some file in \fB/dev/fd\fP. The name of this file is
passed as an argument to the current command as the result of the
expansion. If the \fB>(\fP\fIlist\^\fP\fB)\fP form is used, writing to
the file will provide input for \fIlist\fP. If the
\fB<(\fP\fIlist\^\fP\fB)\fP form is used, the file passed as an
argument should be read to obtain the output of \fIlist\fP.
.PP
When available, process substitution is performed
simultaneously with parameter and variable expansion,
command substitution,
and arithmetic expansion.
.SS Word Splitting
.PP
The shell scans the results of
parameter expansion,
command substitution,
and
arithmetic expansion
that did not occur within double quotes for
.IR "word splitting" .
.PP
The shell treats each character of
.SM
.B IFS
as a delimiter, and splits the results of the other
expansions into words on these characters. If
.SM
.B IFS
is unset, or its
value is exactly
.BR <space><tab><newline> ,
the default, then
any sequence of
.SM
.B IFS
characters serves to delimit words. If
.SM
.B IFS
has a value other than the default, then sequences of
the whitespace characters
.B space
and
.B tab
are ignored at the beginning and end of the
word, as long as the whitespace character is in the
value of
.SM
.BR IFS
(an
.SM
.B IFS
whitespace character).
Any character in
.SM
.B IFS
that is not
.SM
.B IFS
whitespace, along with any adjacent
.SM
.B IFS
whitespace characters, delimits a field.
A sequence of
.SM
.B IFS
whitespace characters is also treated as a delimiter.
If the value of
.SM
.B IFS
is null, no word splitting occurs.
.PP
Explicit null arguments (\^"\^"\fP or \^\f3\(aq\^\(aq\fP\^) are retained.
Unquoted implicit null arguments, resulting from the expansion of
parameters that have no values, are removed.
If a parameter with no value is expanded within double quotes, a
null argument results and is retained.
.PP
Note that if no expansion occurs, no splitting
is performed.
.SS Pathname Expansion
.PP
After word splitting,
unless the
.B \-f
option has been set,
.B bash
scans each word for the characters
.BR * ,
.BR ? ,
and
.BR [.
If one of these characters appears, then the word is
regarded as a
.IR pattern ,
and replaced with an alphabetically sorted list of
file names matching the pattern.
If no matching file names are found,
and the shell option
.B nullglob
is disabled, the word is left unchanged.
If the
.B nullglob
option is set, and no matches are found,
the word is removed.
If the
.B failglob
shell option is set, and no matches are found, an error message
is printed and the command is not executed.
If the shell option
.B nocaseglob
is enabled, the match is performed without regard to the case
of alphabetic characters.
When a pattern is used for pathname expansion,
the character

.B ``.''
at the start of a name or immediately following a slash
must be matched explicitly, unless the shell option
.B dotglob
is set.
When matching a pathname, the slash character must always be
matched explicitly.
In other cases, the
.B ``.''
character is not treated specially.
See the description of
.B shopt
below under
.SM
.B SHELL BUILTIN COMMANDS
for a description of the
.BR nocaseglob ,
.BR nullglob ,
.BR failglob ,
and
.B dotglob
shell options.
.PP
The
.SM
.B GLOBIGNORE
shell variable may be used to restrict the set of file names matching a
.IR pattern .
If
.SM
.B GLOBIGNORE
is set, each matching file name that also matches one of the patterns in
.SM
.B GLOBIGNORE
is removed from the list of matches.
The file names
.B ``.''
and
.B ``..''
are always ignored when
.SM
.B GLOBIGNORE
is set and not null. However, setting
.SM
.B GLOBIGNORE
to a non-null value has the effect of enabling the
.B dotglob
shell option, so all other file names beginning with a
.B ``.''
will match.
To get the old behavior of ignoring file names beginning with a
.BR ``.'' ,
make
.B ``.*''
one of the patterns in
.SM
.BR GLOBIGNORE .
The
.B dotglob
option is disabled when
.SM
.B GLOBIGNORE
is unset.
.PP
\fBPattern Matching\fP
.PP
Any character that appears in a pattern, other than the special pattern
characters described below, matches itself. The NUL character may not
occur in a pattern. A backslash escapes the following character; the
escaping backslash is discarded when matching.
The special pattern characters must be quoted if
they are to be matched literally.
.PP
The special pattern characters have the following meanings:
.PP
.PD 0
.TP
.B *
Matches any string, including the null string.
.TP
.B ?
Matches any single character.
.TP
.B [...]
Matches any one of the enclosed characters. A pair of characters
separated by a hyphen denotes a
\fIrange expression\fP;
any character that sorts between those two characters, inclusive,
using the current locale's collating sequence and character set,
is matched. If the first character following the
.B [
is a
.B !
or a
.B ^
then any character not enclosed is matched.
The sorting order of characters in range expressions is determined by
the current locale and the value of the \fBLC_COLLATE\fP shell variable,
if set.
A
.B \-
may be matched by including it as the first or last character
in the set.
A
.B]
may be matched by including it as the first character
in the set.
.br
.if t .sp 0.5
.if n .sp 1
Within
.B [
and
.BR] ,
\fIcharacter classes\fP can be specified using the syntax
\fB[:\fP\fIclass\fP\fB:]\fP, where \fIclass\fP is one of the
following classes defined in the POSIX standard:
.PP
.RS
.B
.if n alnum alpha ascii blank cntrl digit graph lower print punct space upper word xdigit
.if t alnum   alpha   ascii   blank   cntrl   digit   graph   lower   print   punct   space   upper   word   xdigit

```
.br
A character class matches any character belonging to that class.
The \fBword\fP character class matches letters, digits, and the character _.
.br
.if t .sp 0.5
.if n .sp 1
Within
.B [
and
.BR ] ,
an \fIequivalence class\fP can be specified using the syntax
\fB[=\fP\fIc\fP\fB=]\fP, which matches all characters with the
same collation weight (as defined by the current locale) as
the character \fIc\fP.
.br
.if t .sp 0.5
.if n .sp 1
Within
.B [
and
.BR ] ,
the syntax \fB[.\fP\fIsymbol\fP\fB.]\fP matches the collating symbol
\fIsymbol\fP.
.RE
.PD
.PP
If the \fBextglob\fP shell option is enabled using the \fBshopt\fP
builtin, several extended pattern matching operators are recognized.
In the following description, a \fIpattern-list\fP is a list of one
or more patterns separated by a \fB|\fP.
Composite patterns may be formed using one or more of the following
sub-patterns:
.sp 1
.PD 0
.RS
.TP
\fB?(\fP\^\fIpattern-list\^\fP\fB)\fP
Matches zero or one occurrence of the given patterns
.TP
\fB*(\fP\^\fIpattern-list\^\fP\fB)\fP
Matches zero or more occurrences of the given patterns
.TP
\fB+(\fP\^\fIpattern-list\^\fP\fB)\fP
Matches one or more occurrences of the given patterns
.TP
\fB@(\fP\^\fIpattern-list\^\fP\fB)\fP
Matches one of the given patterns
.TP
\fB!(\fP\^\fIpattern-list\^\fP\fB)\fP
Matches anything except one of the given patterns
.RE
.PD
.SS Quote Removal
.PP
After the preceding expansions, all unquoted occurrences of the
characters
.BR \e ,
.BR \(aq ,
and \^\f3"\fP\^ that did not result from one of the above
expansions are removed.
.SH REDIRECTION
Before a command is executed, its input and output
may be
.I redirected
using a special notation interpreted by the shell.
Redirection may also be used to open and close files for the
current shell execution environment.  The following redirection
operators may precede or appear anywhere within a
.I simple command
or may follow a
.IR command .
Redirections are processed in the order they appear, from
left to right.
.PP
In the following descriptions, if the file descriptor number is
omitted, and the first character of the redirection operator is
.BR < ,
the redirection refers to the standard input (file descriptor
0).  If the first character of the redirection operator is
.BR > ,
the redirection refers to the standard output (file descriptor
1).
.PP
The word following the redirection operator in the following
descriptions, unless otherwise noted, is subjected to brace expansion,
tilde expansion, parameter expansion, command substitution, arithmetic
expansion, quote removal, pathname expansion, and word splitting.
If it expands to more than one word,
.B bash
reports an error.
.PP
Note that the order of redirections is significant.  For example,
the command
.RS
.PP
ls \fB>\fP dirlist 2\fB>&\fP1
.RE
.PP
directs both standard output and standard error to the file
.IR dirlist ,
while the command
.RS
.PP
ls 2\fB>&\fP1 \fB>\fP dirlist
.RE
.PP
directs only the standard output to file
.IR dirlist ,
because the standard error was duplicated as standard output
before the standard output was redirected to
.IR dirlist .
.PP
\fBBash\fP handles several filenames specially when they are used in
redirections, as described in the following table:
.RS
.PP
.PD 0
.TP
.B /dev/fd/\fIfd\fP
If \fIfd\fP is a valid integer, file descriptor \fIfd\fP is duplicated.
.TP
.B /dev/stdin
```

```
File descriptor 0 is duplicated.
.TP
.B /dev/stdout
File descriptor 1 is duplicated.
.TP
.B /dev/stderr
File descriptor 2 is duplicated.
.TP
.B /dev/tcp/\fIhost\fP/\fIport\fP
If \fIhost\fP is a valid hostname or Internet address, and \fIport\fP
is an integer port number or service name, \fBbash\fP attempts to open
a TCP connection to the corresponding socket.
.TP
.B /dev/udp/\fIhost\fP/\fIport\fP
If \fIhost\fP is a valid hostname or Internet address, and \fIport\fP
is an integer port number or service name, \fBbash\fP attempts to open
a UDP connection to the corresponding socket.
.PD
.RE
.PP
A failure to open or create a file causes the redirection to fail.
.PP
Redirections using file descriptors greater than 9 should be used with
care, as they may conflict with file descriptors the shell uses
internally.
.SS Redirecting Input
.PP
Redirection of input causes the file whose name results from
the expansion of
.I word
to be opened for reading on file descriptor
.IR n ,
or the standard input (file descriptor 0) if
.I n
is not specified.
.PP
The general format for redirecting input is:
.RS
.PP
[\fIn\fP]\fB<\fP\fIword\fP
.RE
.SS Redirecting Output
.PP
Redirection of output causes the file whose name results from
the expansion of
.I word
to be opened for writing on file descriptor
.IR n ,
or the standard output (file descriptor 1) if
.I n
is not specified.  If the file does not exist it is created;
if it does exist it is truncated to zero size.
.PP
The general format for redirecting output is:
.RS
.PP
[\fIn\fP]\fB>\fP\fIword\fP
.RE
.PP
If the redirection operator is
.BR > ,
and the
.B noclobber
option to the
.B set
builtin has been enabled, the redirection will fail if the file
whose name results from the expansion of \fIword\fP exists and is
a regular file.
If the redirection operator is
.BR >| ,
or the redirection operator is
.B >
and the
.B noclobber
option to the
.B set
builtin command is not enabled, the redirection is attempted even
if the file named by \fIword\fP exists.
.SS Appending Redirected Output
.PP
Redirection of output in this fashion
causes the file whose name results from
the expansion of
.I word
to be opened for appending on file descriptor
.IR n ,
or the standard output (file descriptor 1) if
.I n
is not specified.  If the file does not exist it is created.
.PP
The general format for appending output is:
.RS
.PP
[\fIn\fP]\fB>>\fP\fIword\fP
.RE
.PP
.SS Redirecting Standard Output and Standard Error
.PP
.B Bash
allows both the
standard output (file descriptor 1) and
the standard error output (file descriptor 2)
to be redirected to the file whose name is the
expansion of
.I word
with this construct.
.PP
There are two formats for redirecting standard output and
standard error:
.RS
.PP
\fB&>\fP\fIword\fP
.RE
and
.RS
\fB>&\fP\fIword\fP
.RE
.PP
Of the two forms, the first is preferred.
This is semantically equivalent to
.RS
.PP
```

```
\fB>\fP\fIword\fP 2\fB>&\fP1
.RE
.SS Here Documents
.PP
This type of redirection instructs the shell to read input from the
current source until a line containing only
.I word
(with no trailing blanks)
is seen.  All of
the lines read up to that point are then used as the standard
input for a command.
.PP
The format of here-documents is:
.RS
.PP
.nf
\fB<<\fP[\fB\-\fP]\fIword\fP
        \fIhere-document\fP
\fIdelimiter\fP
.fi
.RE
.PP
No parameter expansion, command substitution, arithmetic expansion,
or pathname expansion is performed on
.IR word .
If any characters in
.I word
are quoted, the
.I delimiter
is the result of quote removal on
.IR word ,
and the lines in the here-document are not expanded.
If \fIword\fP is unquoted,
all lines of the here-document are subjected to parameter expansion,
command substitution, and arithmetic expansion.  In the latter
case, the character sequence
.B \e<newline>
is ignored, and
.B \e
must be used to quote the characters
.BR \e ,
.BR $ ,
and
.BR ` .
.PP
If the redirection operator is
.BR <<\- ,
then all leading tab characters are stripped from input lines and the
line containing
.IR delimiter .
This allows
here-documents within shell scripts to be indented in a
natural fashion.
.SS "Here Strings"
A variant of here documents, the format is:
.RS
.PP
.nf
\fB<<<\fP\fIword\fP
.fi
.RE
.PP
The \fIword\fP is expanded and supplied to the command on its standard
input.
.SS "Duplicating File Descriptors"
.PP
The redirection operator
.RS
.PP
[\fIn\fP]\fB<&\fP\fIword\fP
.RE
.PP
is used to duplicate input file descriptors.
If
.I word
expands to one or more digits, the file descriptor denoted by
.I n
is made to be a copy of that file descriptor.
If the digits in
.I word
do not specify a file descriptor open for input, a redirection error occurs.
If
.I word
evaluates to
.BR \- ,
file descriptor
.I n
is closed.  If
.I n
is not specified, the standard input (file descriptor 0) is used.
.PP
The operator
.RS
.PP
[\fIn\fP]\fB>&\fP\fIword\fP
.RE
.PP
is used similarly to duplicate output file descriptors.  If
.I n
is not specified, the standard output (file descriptor 1) is used.
If the digits in
.I word
do not specify a file descriptor open for output, a redirection error occurs.
As a special case, if \fIn\fP is omitted, and \fIword\fP does not
expand to one or more digits, the standard output and standard
error are redirected as described previously.
.SS "Moving File Descriptors"
.PP
The redirection operator
.RS
.PP
[\fIn\fP]\fB<&\fP\fIdigit\fP\fB\-\fP
.RE
.PP
moves the file descriptor \fIdigit\fP to file descriptor
.IR n ,
or the standard input (file descriptor 0) if \fIn\fP is not specified.
\fIdigit\fP is closed after being duplicated to \fIn\fP.
.PP
Similarly, the redirection operator
.RS
.PP
```

[\fIn\fP]\fB>&\fP\fIdigit\fP\fB\-\fP
.RE
.PP
moves the file descriptor \fIdigit\fP to file descriptor
.IR n ,
or the standard output (file descriptor 1) if \fIn\fP is not specified.
.SS "Opening File Descriptors for Reading and Writing"
.PP
The redirection operator
.RS
.PP
[\fIn\fP]\fB<>\fP\fIword\fP
.RE
.PP
causes the file whose name is the expansion of
.I word
to be opened for both reading and writing on file descriptor
.IR n ,
or on file descriptor 0 if
.I n
is not specified. If the file does not exist, it is created.
.SH ALIASES
\fIAliases\fP allow a string to be substituted for a word when it is used
as the first word of a simple command.
The shell maintains a list of aliases that may be set and unset with the
.B alias
and
.B unalias
builtin commands (see
.SM
.B SHELL BUILTIN COMMANDS
below).
The first word of each simple command, if unquoted,
is checked to see if it has an
alias. If so, that word is replaced by the text of the alias.
The characters \fB/\fP, \fB$\fP, \fB`\fP, and \fB=\fP and
any of the shell \fImetacharacters\fP or quoting characters
listed above may not appear in an alias name.
The replacement text may contain any valid shell input,
including shell metacharacters.
The first word of the replacement text is tested
for aliases, but a word that is identical to an alias being expanded
is not expanded a second time.
This means that one may alias
.B ls
to
.BR "ls \-F" ,
for instance, and
.B bash
does not try to recursively expand the replacement text.
If the last character of the alias value is a
.IR blank ,
then the next command
word following the alias is also checked for alias expansion.
.PP
Aliases are created and listed with the
.B alias
command, and removed with the
.B unalias
command.
.PP
There is no mechanism for using arguments in the replacement text.
If arguments are needed, a shell function should be used (see
.SM
.B FUNCTIONS
below).
.PP
Aliases are not expanded when the shell is not interactive, unless
the
.B expand_aliases
shell option is set using
.B shopt
(see the description of
.B shopt
under
.SM
\fBSHELL BUILTIN COMMANDS\fP
below).
.PP
The rules concerning the definition and use of aliases are
somewhat confusing.
.B Bash
always reads at least one complete line
of input before executing any
of the commands on that line. Aliases are expanded when a
command is read, not when it is executed. Therefore, an
alias definition appearing on the same line as another
command does not take effect until the next line of input is read.
The commands following the alias definition
on that line are not affected by the new alias.
This behavior is also an issue when functions are executed.
Aliases are expanded when a function definition is read,
not when the function is executed, because a function definition
is itself a compound command. As a consequence, aliases
defined in a function are not available until after that
function is executed. To be safe, always put
alias definitions on a separate line, and do not use
.B alias
in compound commands.
.PP
For almost every purpose, aliases are superseded by
shell functions.
.SH FUNCTIONS
A shell function, defined as described above under
.SM
.BR "SHELL GRAMMAR" ,
stores a series of commands for later execution.
When the name of a shell function is used as a simple command name,
the list of commands associated with that function name is executed.
Functions are executed in the context of the
current shell; no new process is created to interpret
them (contrast this with the execution of a shell script).
When a function is executed, the arguments to the
function become the positional parameters
during its execution.
The special parameter
.B #
is updated to reflect the change. Special parameter 0
is unchanged.
The first element of the
.SM
.B FUNCNAME

variable is set to the name of the function while the function
is executing.
All other aspects of the shell execution
environment are identical between a function and its caller
with the exception that the
.SM
.B DEBUG
and
.B RETURN
traps (see the description of the
.B trap
builtin under
.SM
.B SHELL BUILTIN COMMANDS
below) are not inherited unless the function has been given the
\fBtrace\fP attribute (see the description of the
.SM
.B declare
builtin below) or the
\fB\-o functrace\fP shell option has been enabled with
the \fBset\fP builtin
(in which case all functions inherit the \fBDEBUG\fP and \fBRETURN\fP traps).
.PP
Variables local to the function may be declared with the
.B local
builtin command. Ordinarily, variables and their values
are shared between the function and its caller.
.PP
If the builtin command
.B return
is executed in a function, the function completes and
execution resumes with the next command after the function
call.
Any command associated with the \fBRETURN\fP trap is executed
before execution resumes.
When a function completes, the values of the
positional parameters and the special parameter
.B #
are restored to the values they had prior to the function's
execution.
.PP
Function names and definitions may be listed with the
.B \-f
option to the
.B declare
or
.B typeset
builtin commands. The
.B \-F
option to
.B declare
or
.B typeset
will list the function names only
(and optionally the source file and line number, if the \fBextdebug\fP
shell option is enabled).
Functions may be exported so that subshells
automatically have them defined with the
.B \-f
option to the
.B export
builtin.
A function definition may be deleted using the \fB\-f\fP option to
the
.B unset
builtin.
Note that shell functions and variables with the same name may result
in multiple identically-named entries in the environment passed to the
shell's children.
Care should be taken in cases where this may cause a problem.
.PP
Functions may be recursive. No limit is imposed on the number
of recursive calls.
.SH "ARITHMETIC EVALUATION"
The shell allows arithmetic expressions to be evaluated, under
certain circumstances (see the \fBlet\fP and \fBdeclare\fP builtin
commands and \fBArithmetic Expansion\fP).
Evaluation is done in fixed-width integers with no check for overflow,
though division by 0 is trapped and flagged as an error.
The operators and their precedence, associativity, and values
are the same as in the C language.
The following list of operators is grouped into levels of
equal-precedence operators.
The levels are listed in order of decreasing precedence.
.PP
.PD 0
.TP
.B \fIid\fP++ \fIid\fP\-\-
variable post-increment and post-decrement
.TP
.B ++\fIid\fP \-\-\fIid\fP
variable pre-increment and pre-decrement
.TP
.B \- +
unary minus and plus
.TP
.B ! ~
logical and bitwise negation
.TP
.B **
exponentiation
.TP
.B * / %
multiplication, division, remainder
.TP
.B + \-
addition, subtraction
.TP
.B << >>
left and right bitwise shifts
.TP
.B <= >= < >
comparison
.TP
.B == !=
equality and inequality
.TP
.B &
bitwise AND
.TP
.B ^
bitwise exclusive OR

.TP
.B |
bitwise OR
.TP
.B &&
logical AND
.TP
.B ||
logical OR
.TP
.B \fIexpr\fP?\fIexpr\fP:\fIexpr\fP
conditional operator
.TP
.B = *= /= %= += \-= <<= >>= &= ^= |=
assignment
.TP
.B \fIexpr1\fP , \fIexpr2\fP
comma
.PD
.PP
Shell variables are allowed as operands; parameter expansion is
performed before the expression is evaluated.
Within an expression, shell variables may also be referenced by name
without using the parameter expansion syntax.
A shell variable that is null or unset evaluates to 0 when referenced
by name without using the parameter expansion syntax.
The value of a variable is evaluated as an arithmetic expression
when it is referenced, or when a variable which has been given the
\fIinteger\fP attribute using \fBdeclare -i\fP is assigned a value.
A null value evaluates to 0.
A shell variable need not have its integer attribute
turned on to be used in an expression.
.PP
Constants with a leading 0 are interpreted as octal numbers.
A leading 0x or 0X denotes hexadecimal.
Otherwise, numbers take the form [\fIbase#\fP]n, where \fIbase\fP
is a decimal number between 2 and 64 representing the arithmetic
base, and \fIn\fP is a number in that base.
If \fIbase#\fP is omitted, then base 10 is used.
The digits greater than 9 are represented by the lowercase letters,
the uppercase letters, @, and _, in that order.
If \fIbase\fP is less than or equal to 36, lowercase and uppercase
letters may be used interchangeably to represent numbers between 10
and 35.
.PP
Operators are evaluated in order of precedence. Sub-expressions in
parentheses are evaluated first and may override the precedence
rules above.
.SH "CONDITIONAL EXPRESSIONS"
Conditional expressions are used by the \fB[[\fP compound command and
the \fBtest\fP and \fB[\fP builtin commands to test file attributes
and perform string and arithmetic comparisons.
Expressions are formed from the following unary or binary primaries.
If any \fIfile\fP argument to one of the primaries is of the form
\fI/dev/fd/n\fP, then file descriptor \fIn\fP is checked.
If the \fIfile\fP argument to one of the primaries is one of
\fI/dev/stdin\fP, \fI/dev/stdout\fP, or \fI/dev/stderr\fP, file
descriptor 0, 1, or 2, respectively, is checked.
.PP
Unless otherwise specified, primaries that operate on files follow symbolic
links and operate on the target of the link, rather than the link itself.
.sp 1
.PD 0
.TP
.B \-a \fIfile\fP
True if \fIfile\fP exists.
.TP
.B \-b \fIfile\fP
True if \fIfile\fP exists and is a block special file.
.TP
.B \-c \fIfile\fP
True if \fIfile\fP exists and is a character special file.
.TP
.B \-d \fIfile\fP
True if \fIfile\fP exists and is a directory.
.TP
.B \-e \fIfile\fP
True if \fIfile\fP exists.
.TP
.B \-f \fIfile\fP
True if \fIfile\fP exists and is a regular file.
.TP
.B \-g \fIfile\fP
True if \fIfile\fP exists and is set-group-id.
.TP
.B \-h \fIfile\fP
True if \fIfile\fP exists and is a symbolic link.
.TP
.B \-k \fIfile\fP
True if \fIfile\fP exists and its ``sticky'' bit is set.
.TP
.B \-p \fIfile\fP
True if \fIfile\fP exists and is a named pipe (FIFO).
.TP
.B \-r \fIfile\fP
True if \fIfile\fP exists and is readable.
.TP
.B \-s \fIfile\fP
True if \fIfile\fP exists and has a size greater than zero.
.TP
.B \-t \fIfd\fP
True if file descriptor
.I fd
is open and refers to a terminal.
.TP
.B \-u \fIfile\fP
True if \fIfile\fP exists and its set-user-id bit is set.
.TP
.B \-w \fIfile\fP
True if \fIfile\fP exists and is writable.
.TP
.B \-x \fIfile\fP
True if \fIfile\fP exists and is executable.
.TP
.B \-O \fIfile\fP
True if \fIfile\fP exists and is owned by the effective user id.
.TP
.B \-G \fIfile\fP
True if \fIfile\fP exists and is owned by the effective group id.
.TP
.B \-L \fIfile\fP
True if \fIfile\fP exists and is a symbolic link.

```
.TP
.B \-S \fIfile\fP
True if \fIfile\fP exists and is a socket.
.TP
.B \-N \fIfile\fP
True if \fIfile\fP exists and has been modified since it was last read.
.TP
\fIfile1\fP \-\fBnt\fP \fIfile2\fP
True if \fIfile1\fP is newer (according to modification date) than \fIfile2\fP,
or if \fIfile1\fP exists and \fPfile2\fP does not.
.TP
\fIfile1\fP \-\fBot\fP \fIfile2\fP
True if \fIfile1\fP is older than \fIfile2\fP, or if \fIfile2\fP exists
and \fIfile1\fP does not.
.TP
\fIfile1\fP \fB\-ef\fP \fIfile2\fP
True if \fIfile1\fP and \fIfile2\fP refer to the same device and
inode numbers.
.TP
.B \-o \fIoptname\fP
True if shell option
.I optname
is enabled.
See the list of options under the description of the
.B \-o
option to the
.B set
builtin below.
.TP
.B \-z \fIstring\fP
True if the length of \fIstring\fP is zero.
.TP
\fIstring\fP
.PD 0
.TP
.B \-n \fIstring\fP
.PD
True if the length of
.I string
is non-zero.
.TP
\fIstring1\fP \fB==\fP \fIstring2\fP
True if the strings are equal.  \fB=\fP may be used in place of
\fB==\fP for strict POSIX compliance.
.TP
\fIstring1\fP \fB!=\fP \fIstring2\fP
True if the strings are not equal.
.TP
\fIstring1\fP \fB<\fP \fIstring2\fP
True if \fIstring1\fP sorts before \fIstring2\fP lexicographically
in the current locale.
.TP
\fIstring1\fP \fB>\fP \fIstring2\fP
True if \fIstring1\fP sorts after \fIstring2\fP lexicographically
in the current locale.
.TP
.I \fIarg1\fP \fBOP\fP \fIarg2\fP
.SM
.B OP
is one of
.BR \-eq ,
.BR \-ne ,
.BR \-lt ,
.BR \-le ,
.BR \-gt ,
or
.BR \-ge .
These arithmetic binary operators return true if \fIarg1\fP
is equal to, not equal to, less than, less than or equal to,
greater than, or greater than or equal to \fIarg2\fP, respectively.
.I Arg1
and
.I arg2
may be positive or negative integers.
.PD
.SH "SIMPLE COMMAND EXPANSION"
When a simple command is executed, the shell performs the following
expansions, assignments, and redirections, from left to right.
.IP 1.
The words that the parser has marked as variable assignments (those
preceding the command name) and redirections are saved for later
processing.
.IP 2.
The words that are not variable assignments or redirections are
expanded.  If any words remain after expansion, the first word
is taken to be the name of the command and the remaining words are
the arguments.
.IP 3.
Redirections are performed as described above under
.SM
.BR REDIRECTION .
.IP 4.
The text after the \fB=\fP in each variable assignment undergoes tilde
expansion, parameter expansion, command substitution, arithmetic expansion,
and quote removal before being assigned to the variable.
.PP
If no command name results, the variable assignments affect the current
shell environment.  Otherwise, the variables are added to the environment
of the executed command and do not affect the current shell environment.
If any of the assignments attempts to assign a value to a readonly variable,
an error occurs, and the command exits with a non-zero status.
.PP
If no command name results, redirections are performed, but do not
affect the current shell environment.  A redirection error causes the
command to exit with a non-zero status.
.PP
If there is a command name left after expansion, execution proceeds as
described below.  Otherwise, the command exits.  If one of the expansions
contained a command substitution, the exit status of the command is
the exit status of the last command substitution performed.  If there
were no command substitutions, the command exits with a status of zero.
.SH "COMMAND EXECUTION"
After a command has been split into words, if it results in a
simple command and an optional list of arguments, the following
actions are taken.
.PP
If the command name contains no slashes, the shell attempts to
locate it.  If there exists a shell function by that name, that
function is invoked as described above in
.SM
.BR FUNCTIONS .
If the name does not match a function, the shell searches for
```

it in the list of shell builtins. If a match is found, that
builtin is invoked.
.PP
If the name is neither a shell function nor a builtin,
and contains no slashes,
.B bash
searches each element of the
.SM
.B PATH
for a directory containing an executable file by that name.
.B Bash
uses a hash table to remember the full pathnames of executable
files (see
.B hash
under
.SM
.B "SHELL BUILTIN COMMANDS"
below).
A full search of the directories in
.SM
.B PATH
is performed only if the command is not found in the hash table.
If the search is unsuccessful, the shell prints an error
message and returns an exit status of 127.
.PP
If the search is successful, or if the command name contains
one or more slashes, the shell executes the named program in a
separate execution environment.
Argument 0 is set to the name given, and the remaining arguments
to the command are set to the arguments given, if any.
.PP
If this execution fails because the file is not in executable
format, and the file is not a directory, it is assumed to be
a \fIshell script\fP, a file
containing shell commands. A subshell is spawned to execute
it. This subshell reinitializes itself, so
that the effect is as if a new shell had been invoked
to handle the script, with the exception that the locations of
commands remembered by the parent (see
.B hash
below under
.SM
\fBSHELL BUILTIN COMMANDS\fP)
are retained by the child.
.PP
If the program is a file beginning with
.BR #! ,
the remainder of the first line specifies an interpreter
for the program. The shell executes the
specified interpreter on operating systems that do not
handle this executable format themselves. The arguments to the
interpreter consist of a single optional argument following the
interpreter name on the first line of the program, followed
by the name of the program, followed by the command
arguments, if any.
.SH COMMAND EXECUTION ENVIRONMENT
The shell has an \fIexecution environment\fP, which consists of the
following:
.sp 1
.IP \(bu
open files inherited by the shell at invocation, as modified by
redirections supplied to the \fBexec\fP builtin
.IP \(bu
the current working directory as set by \fBcd\fP, \fBpushd\fP, or
\fBpopd\fP, or inherited by the shell at invocation
.IP \(bu
the file creation mode mask as set by \fBumask\fP or inherited from
the shell's parent
.IP \(bu
current traps set by \fBtrap\fP
.IP \(bu
shell parameters that are set by variable assignment or with \fBset\fP
or inherited from the shell's parent in the environment
.IP \(bu
shell functions defined during execution or inherited from the shell's
parent in the environment
.IP \(bu
options enabled at invocation (either by default or with command-line
arguments) or by \fBset\fP
.IP \(bu
options enabled by \fBshopt\fP
.IP \(bu
shell aliases defined with \fBalias\fP
.IP \(bu
various process IDs, including those of background jobs, the value
of \fB$$\fP, and the value of \fB$PPID\fP
.PP
When a simple command other than a builtin or shell function
is to be executed, it
is invoked in a separate execution environment that consists of
the following. Unless otherwise noted, the values are inherited
from the shell.
.sp 1
.IP \(bu
the shell's open files, plus any modifications and additions specified
by redirections to the command
.IP \(bu
the current working directory
.IP \(bu
the file creation mode mask
.IP \(bu
shell variables and functions marked for export, along with variables
exported for the command, passed in the environment
.IP \(bu
traps caught by the shell are reset to the values inherited from the
shell's parent, and traps ignored by the shell are ignored
.PP
A command invoked in this separate environment cannot affect the
shell's execution environment.
.PP
Command substitution, commands grouped with parentheses,
and asynchronous commands are invoked in a
subshell environment that is a duplicate of the shell environment,
except that traps caught by the shell are reset to the values
that the shell inherited from its parent at invocation. Builtin
commands that are invoked as part of a pipeline are also executed in a
subshell environment. Changes made to the subshell environment
cannot affect the shell's execution environment.
.PP
If a command is followed by a \fB&\fP and job control is not active, the
default standard input for the command is the empty file \fI/dev/null\fP.
Otherwise, the invoked command inherits the file descriptors of the calling

```
shell as modified by redirections.
.SH ENVIRONMENT
When a program is invoked it is given an array of strings
called the
.IR environment .
This is a list of
\fIname\fP\-\fIvalue\fP pairs, of the form
.IR "name\fR=\fPvalue" .
.PP
The shell provides several ways to manipulate the environment.
On invocation, the shell scans its own environment and
creates a parameter for each name found, automatically marking
it for
.I export
to child processes.  Executed commands inherit the environment.
The
.B export
and
.B declare \-x
commands allow parameters and functions to be added to and
deleted from the environment.  If the value of a parameter
in the environment is modified, the new value becomes part
of the environment, replacing the old.  The environment
inherited by any executed command consists of the shell's
initial environment, whose values may be modified in the shell,
less any pairs removed by the
.B unset
command, plus any additions via the
.B export
and
.B declare \-x
commands.
.PP
The environment for any
.I simple command
or function may be augmented temporarily by prefixing it with
parameter assignments, as described above in
.SM
.BR PARAMETERS .
These assignment statements affect only the environment seen
by that command.
.PP
If the
.B \-k
option is set (see the
.B set
builtin command below), then
.I all
parameter assignments are placed in the environment for a command,
not just those that precede the command name.
.PP
When
.B bash
invokes an external command, the variable
.B _
is set to the full file name of the command and passed to that
command in its environment.
.SH "EXIT STATUS"
For the shell's purposes, a command which exits with a
zero exit status has succeeded.  An exit status of zero
indicates success.  A non-zero exit status indicates failure.
When a command terminates on a fatal signal \fIN\fP, \fBbash\fP uses
the value of 128+\fIN\fP as the exit status.
.PP
If a command is not found, the child process created to
execute it returns a status of 127.  If a command is found
but is not executable, the return status is 126.
.PP
If a command fails because of an error during expansion or redirection,
the exit status is greater than zero.
.PP
Shell builtin commands return a status of 0 (\fItrue\fP) if
successful, and non-zero (\fIfalse\fP) if an error occurs
while they execute.
All builtins return an exit status of 2 to indicate incorrect usage.
.PP
\fBBash\fP itself returns the exit status of the last command
executed, unless a syntax error occurs, in which case it exits
with a non-zero value.  See also the \fBexit\fP builtin
command below.
.SH SIGNALS
When \fBbash\fP is interactive, in the absence of any traps, it ignores
.SM
.B SIGTERM
(so that \fBkill 0\fP does not kill an interactive shell),
and
.SM
.B SIGINT
is caught and handled (so that the \fBwait\fP builtin is interruptible).
In all cases, \fBbash\fP ignores
.SM
.BR SIGQUIT .
If job control is in effect,
.B bash
ignores
.SM
.BR SIGTTIN ,
.SM
.BR SIGTTOU ,
and
.SM
.BR SIGTSTP .
.PP
Non-builtin commands run by \fBbash\fP have signal handlers
set to the values inherited by the shell from its parent.
When job control is not in effect, asynchronous commands
ignore
.SM
.B SIGINT
and
.SM
.B SIGQUIT
in addition to these inherited handlers.
Commands run as a result of command substitution ignore the
keyboard-generated job control signals
.SM
.BR SIGTTIN ,
.SM
.BR SIGTTOU ,
and
.SM
.BR SIGTSTP .
```

.PP
The shell exits by default upon receipt of a
.SM
.BR SIGHUP .
Before exiting, an interactive shell resends the
.SM
.B SIGHUP
to all jobs, running or stopped.
Stopped jobs are sent
.SM
.B SIGCONT
to ensure that they receive the
.SM
.BR SIGHUP .
To prevent the shell from
sending the signal to a particular job, it should be removed from the
jobs table with the
.B disown
builtin (see
.SM
.B "SHELL BUILTIN COMMANDS"
below) or marked
to not receive
.SM
.B SIGHUP
using
.BR "disown \-h" .
.PP
If the
.B huponexit
shell option has been set with
.BR shopt ,
.B bash
sends a
.SM
.B SIGHUP
to all jobs when an interactive login shell exits.
.PP
If \fBbash\fP is waiting for a command to complete and receives a signal
for which a trap has been set, the trap will not be executed until
the command completes.
When \fBbash\fP is waiting for an asynchronous command via the \fBwait\fP
builtin, the reception of a signal for which a trap has been set will
cause the \fBwait\fP builtin to return immediately with an exit status
greater than 128, immediately after which the trap is executed.
.SH "JOB CONTROL"
.I Job control
refers to the ability to selectively stop (\fIsuspend\fP)
the execution of processes and continue (\fIresume\fP)
their execution at a later point. A user typically employs
this facility via an interactive interface supplied jointly
by the system's terminal driver and
.BR bash .
.PP
The shell associates a
.I job
with each pipeline. It keeps a table of currently executing
jobs, which may be listed with the
.B jobs
command. When
.B bash
starts a job asynchronously (in the
.IR background),
it prints a line that looks like:
.RS
.PP
[1] 25647
.RE
.PP
indicating that this job is job number 1 and that the process ID
of the last process in the pipeline associated with this job is 25647.
All of the processes in a single pipeline are members of the same job.
.B Bash
uses the
.I job
abstraction as the basis for job control.
.PP
To facilitate the implementation of the user interface to job
control, the operating system maintains the notion of a \fIcurrent terminal
process group ID\fP. Members of this process group (processes whose
process group ID is equal to the current terminal process group ID)
receive keyboard-generated signals such as
.SM
.BR SIGINT .
These processes are said to be in the
.IR foreground .
.I Background
processes are those whose process group ID differs from the terminal's;
such processes are immune to keyboard-generated signals.
Only foreground processes are allowed to read from or write to the
terminal. Background processes which attempt to read from (write to) the
terminal are sent a
.SM
.B SIGTTIN (SIGTTOU)
signal by the terminal driver,
which, unless caught, suspends the process.
.PP
If the operating system on which
.B bash
is running supports
job control,
.B bash
contains facilities to use it.
Typing the
.I suspend
character (typically
.BR ^Z ,
Control-Z) while a process is running
causes that process to be stopped and returns control to
.BR bash .
Typing the
.I "delayed suspend"
character (typically
.BR ^Y ,
Control-Y) causes the process to be stopped when it
attempts to read input from the terminal, and control to
be returned to
.BR bash .
The user may then manipulate the state of this job, using the
.B bg
command to continue it in the background, the
.B fg

command to continue it in the foreground, or
the
.B kill
command to kill it. A \fB^Z\fP takes effect immediately,
and has the additional side effect of causing pending output
and typeahead to be discarded.
.PP
There are a number of ways to refer to a job in the shell.
The character
.B %
introduces a job name. Job number
.I n
may be referred to as
.BR %n .
A job may also be referred to using a prefix of the name used to
start it, or using a substring that appears in its command line.
For example,
.B %ce
refers to a stopped
.B ce
job. If a prefix matches more than one job,
.B bash
reports an error. Using
.BR %?ce ,
on the other hand, refers to any job containing the string
.B ce
in its command line. If the substring matches more than one job,
.B bash
reports an error. The symbols
.B %%
and
.B %+
refer to the shell's notion of the
.IR "current job" ,
which is the last job stopped while it was in
the foreground or started in the background.
The
.I "previous job"
may be referenced using
.BR %\- .
In output pertaining to jobs (e.g., the output of the
.B jobs
command), the current job is always flagged with a
.BR + ,
and the previous job with a
.BR \- .
A single % (with no accompanying job specification) also refers to the
current job.
.PP
Simply naming a job can be used to bring it into the
foreground:
.B %1
is a synonym for
\fB``fg %1''\fP,
bringing job 1 from the background into the foreground.
Similarly,
.B ``%1 &''
resumes job 1 in the background, equivalent to
\fB``bg %1''\fP.
.PP
The shell learns immediately whenever a job changes state.
Normally,
.B bash
waits until it is about to print a prompt before reporting
changes in a job's status so as to not interrupt
any other output. If the
.B \-b
option to the
.B set
builtin command
is enabled,
.B bash
reports such changes immediately.
Any trap on
.SM
.B SIGCHLD
is executed for each child that exits.
.PP
If an attempt to exit
.B bash
is made while jobs are stopped, the shell prints a warning message. The
.B jobs
command may then be used to inspect their status.
If a second attempt to exit is made without an intervening command,
the shell does not print another warning, and the stopped
jobs are terminated.
.SH PROMPTING
When executing interactively,
.B bash
displays the primary prompt
.SM
.B PS1
when it is ready to read a command, and the secondary prompt
.SM
.B PS2
when it needs more input to complete a command.
.B Bash
allows these prompt strings to be customized by inserting a number of
backslash-escaped special characters that are decoded as follows:
.RS
.PD 0
.TP
.B \ea
an ASCII bell character (07)
.TP
.B \ed
the date in "Weekday Month Date" format (e.g., "Tue May 26")
.TP
.B \eD{\fIformat\fP}
the \fIformat\fP is passed to \fIstrftime\fP(3) and the result is inserted
into the prompt string; an empty \fIformat\fP results in a locale-specific
time representation. The braces are required
.TP
.B \ee
an ASCII escape character (033)
.TP
.B \eh
the hostname up to the first `.'
.TP
.B \eH
the hostname
.TP

.B \ej
the number of jobs currently managed by the shell
.TP
.B \el
the basename of the shell's terminal device name
.TP
.B \en
newline
.TP
.B \er
carriage return
.TP
.B \es
the name of the shell, the basename of
.B $0
(the portion following the final slash)
.TP
.B \et
the current time in 24-hour HH:MM:SS format
.TP
.B \eT
the current time in 12-hour HH:MM:SS format
.TP
.B \e@
the current time in 12-hour am/pm format
.TP
.B \eA
the current time in 24-hour HH:MM format
.TP
.B \eu
the username of the current user
.TP
.B \ev
the version of \fBbash\fP (e.g., 2.00)
.TP
.B \eV
the release of \fBbash\fP, version + patch level (e.g., 2.00.0)
.TP
.B \ew
the current working directory, with \fB$HOME\fP abbreviated with a tilde
.TP
.B \eW
the basename of the current working directory, with \fB$HOME\fP
abbreviated with a tilde
.TP
.B \e!
the history number of this command
.TP
.B \e#
the command number of this command
.TP
.B \e$
if the effective UID is 0, a
.BR # ,
otherwise a
.B $
.TP
.B \e\fInnn\fP
the character corresponding to the octal number \fInnn\fP
.TP
.B \e\e
a backslash
.TP
.B \e[
begin a sequence of non-printing characters, which could be used to
embed a terminal control sequence into the prompt
.TP
.B \e]
end a sequence of non-printing characters
.PD
.RE
.PP
The command number and the history number are usually different:
the history number of a command is its position in the history
list, which may include commands restored from the history file
(see
.SM
.B HISTORY
below), while the command number is the position in the sequence
of commands executed during the current shell session.
After the string is decoded, it is expanded via
parameter expansion, command substitution, arithmetic
expansion, and quote removal, subject to the value of the
.B promptvars
shell option (see the description of the
.B shopt
command under
.SM
.B "SHELL BUILTIN COMMANDS"
below).
.SH READLINE
This is the library that handles reading input when using an interactive
shell, unless the
.B \-\-noediting
option is given at shell invocation.
By default, the line editing commands are similar to those of emacs.
A vi-style line editing interface is also available.
To turn off line editing after the shell is running, use the
.B +o emacs
or
.B +o vi
options to the
.B set
builtin (see
.SM
.B SHELL BUILTIN COMMANDS
below).
.SS "Readline Notation"
.PP
In this section, the emacs-style notation is used to denote
keystrokes. Control keys are denoted by C\-\fIkey\fR, e.g., C\-n
means Control\-N. Similarly,
.I meta
keys are denoted by M\-\fIkey\fR, so M\-x means Meta\-X. (On keyboards
without a
.I meta
key, M\-\fIx\fP means ESC \fIx\fP, i.e., press the Escape key
then the
.I x
key. This makes ESC the \fImeta prefix\fP.
The combination M\-C\-\fIx\fP means ESC\-Control\-\fIx\fP,
or press the Escape key

then hold the Control key while pressing the
.I x
key.)
.PP
Readline commands may be given numeric
.IR arguments ,
which normally act as a repeat count.
Sometimes, however, it is the sign of the argument that is significant.
Passing a negative argument to a command that acts in the forward
direction (e.g., \fBkill\-line\fP) causes that command to act in a
backward direction.
Commands whose behavior with arguments deviates from this are noted
below.
.PP
When a command is described as \fIkilling\fP text, the text
deleted is saved for possible future retrieval
(\fIyanking\fP). The killed text is saved in a
\fIkill ring\fP. Consecutive kills cause the text to be
accumulated into one unit, which can be yanked all at once.
Commands which do not kill text separate the chunks of text
on the kill ring.
.SS "Readline Initialization"
.PP
Readline is customized by putting commands in an initialization
file (the \fIinputrc\fP file).
The name of this file is taken from the value of the
.SM
.B INPUTRC
variable. If that variable is unset, the default is
.IR ~/.inputrc .
When a program which uses the readline library starts up, the
initialization file is read, and the key bindings and variables
are set.
There are only a few basic constructs allowed in the
readline initialization file.
Blank lines are ignored.
Lines beginning with a \fB#\fP are comments.
Lines beginning with a \fB$\fP indicate conditional constructs.
Other lines denote key bindings and variable settings.
.PP
The default key-bindings may be changed with an
.I inputrc
file.
Other programs that use this library may add their own commands
and bindings.
.PP
For example, placing
.RS
.PP
M\-Control\-u: universal\-argument
.RE
or
.RS
C\-Meta\-u: universal\-argument
.RE
into the
.I inputrc
would make M\-C\-u execute the readline command
.IR universal\-argument .
.PP
The following symbolic character names are recognized:
.IR RUBOUT ,
.IR DEL ,
.IR ESC ,
.IR LFD ,
.IR NEWLINE ,
.IR RET ,
.IR RETURN ,
.IR SPC ,
.IR SPACE ,
and
.IR TAB .
.PP
In addition to command names, readline allows keys to be bound
to a string that is inserted when the key is pressed (a \fImacro\fP).
.SS "Readline Key Bindings"
.PP
The syntax for controlling key bindings in the
.I inputrc
file is simple. All that is required is the name of the
command or the text of a macro and a key sequence to which
it should be bound. The name may be specified in one of two ways:
as a symbolic key name, possibly with \fIMeta\-\fP or \fIControl\-\fP
prefixes, or as a key sequence.
.PP
When using the form \fBkeyname\fP:\^\fIfunction\-name\fP or \fImacro\fP,
.I keyname
is the name of a key spelled out in English. For example:
.sp
.RS
Control-u: universal\-argument
.br
Meta-Rubout: backward-kill-word
.br
Control-o: "> output"
.RE
.LP
In the above example,
.I C\-u
is bound to the function
.BR universal\-argument ,
.I M\-DEL
is bound to the function
.BR backward\-kill\-word ,
and
.I C\-o
is bound to run the macro
expressed on the right hand side (that is, to insert the text
.if t \f(CW> output\fP
.if n ``> output''
into the line).
.PP
In the second form, \fB"keyseq"\fP:\^\fIfunction\-name\fP or \fImacro\fP,
.B keyseq
differs from
.B keyname
above in that strings denoting
an entire key sequence may be specified by placing the sequence
within double quotes. Some GNU Emacs style key escapes can be
used, as in the following example, but the symbolic character names
are not recognized.
.sp

.RS
"\eC\-u": universal\-argument
.br
"\eC\-x\eC\-r": re\-read\-init\-file
.br
"\ee[11~": "Function Key 1"
.RE
.PP
In this example,
.I C\-u
is again bound to the function
.BR universal\-argument .
.I "C\-x C\-r"
is bound to the function
.BR re\-read\-init\-file ,
and
.I "ESC [1 1 ~"
is bound to insert the text
.if t \f(CWFunction Key 1\fP.
.if n ``Function Key 1''.
.PP
The full set of GNU Emacs style escape sequences is
.RS
.PD 0
.TP
.B \eC\-
control prefix
.TP
.B \eM\-
meta prefix
.TP
.B \ee
an escape character
.TP
.B \e\e
backslash
.TP
.B \e"
literal "
.TP
.B \e\(aq
literal \(aq
.RE
.PD
.PP
In addition to the GNU Emacs style escape sequences, a second
set of backslash escapes is available:
.RS
.PD 0
.TP
.B \ea
alert (bell)
.TP
.B \eb
backspace
.TP
.B \ed
delete
.TP
.B \ef
form feed
.TP
.B \en
newline
.TP
.B \er
carriage return
.TP
.B \et
horizontal tab
.TP
.B \ev
vertical tab
.TP
.B \e\fInnn\fP
the eight-bit character whose value is the octal value \fInnn\fP
(one to three digits)
.TP
.B \ex\fIHH\fP
the eight-bit character whose value is the hexadecimal value \fIHH\fP
(one or two hex digits)
.RE
.PD
.PP
When entering the text of a macro, single or double quotes must
be used to indicate a macro definition.
Unquoted text is assumed to be a function name.
In the macro body, the backslash escapes described above are expanded.
Backslash will quote any other character in the macro text,
including " and \(aq.
.PP
.B Bash
allows the current readline key bindings to be displayed or modified
with the
.B bind
builtin command. The editing mode may be switched during interactive
use by using the
.B \-o
option to the
.B set
builtin command (see
.SM
.B SHELL BUILTIN COMMANDS
below).
.SS "Readline Variables"
.PP
Readline has variables that can be used to further customize its
behavior. A variable may be set in the
.I inputrc
file with a statement of the form
.RS
.PP
\fBset\fP \fIvariable\-name\fP \fIvalue\fP
.RE
.PP
Except where noted, readline variables can take the values
.B On
or
.B Off
(without regard to case).
Unrecognized variable names are ignored.
When a variable value is read, empty or null values, "on" (case-insensitive),

and "1" are equivalent to \fBOn\fP. All other values are equivalent to
\fBOff\fP.
The variables and their default values are:
.PP
.PD 0
.TP
.B bell\-style (audible)
Controls what happens when readline wants to ring the terminal bell.
If set to \fBnone\fP, readline never rings the bell. If set to
\fBvisible\fP, readline uses a visible bell if one is available.
If set to \fBaudible\fP, readline attempts to ring the terminal's bell.
.TP
.B bind\-tty\-special\-chars (On)
If set to \fBOn\fP, readline attempts to bind the control characters
treated specially by the kernel's terminal driver to their readline
equivalents.
.TP
.B comment\-begin (``#'')
The string that is inserted when the readline
.B insert\-comment
command is executed.
This command is bound to
.B M\-#
in emacs mode and to
.B #
in vi command mode.
.TP
.B completion\-ignore\-case (Off)
If set to \fBOn\fP, readline performs filename matching and completion
in a case\-insensitive fashion.
.TP
.B completion\-query\-items (100)
This determines when the user is queried about viewing
the number of possible completions
generated by the \fBpossible\-completions\fP command.
It may be set to any integer value greater than or equal to
zero. If the number of possible completions is greater than
or equal to the value of this variable, the user is asked whether
or not he wishes to view them; otherwise they are simply listed
on the terminal.
.TP
.B convert\-meta (On)
If set to \fBOn\fP, readline will convert characters with the
eighth bit set to an ASCII key sequence
by stripping the eighth bit and prefixing an
escape character (in effect, using escape as the \fImeta prefix\fP).
.TP
.B disable\-completion (Off)
If set to \fBOn\fP, readline will inhibit word completion. Completion
characters will be inserted into the line as if they had been
mapped to \fBself-insert\fP.
.TP
.B editing\-mode (emacs)
Controls whether readline begins with a set of key bindings similar
to \fIemacs\fP or \fIvi\fP.
.B editing\-mode
can be set to either
.B emacs
or
.BR vi .
.TP
.B enable\-keypad (Off)
When set to \fBOn\fP, readline will try to enable the application
keypad when it is called. Some systems need this to enable the
arrow keys.
.TP
.B expand\-tilde (Off)
If set to \fBon\fP, tilde expansion is performed when readline
attempts word completion.
.TP
.B history\-preserve\-point (Off)
If set to \fBon\fP, the history code attempts to place point at the
same location on each history line retrieved with \fBprevious-history\fP
or \fBnext-history\fP.
.TP
.B horizontal\-scroll\-mode (Off)
When set to \fBOn\fP, makes readline use a single line for display,
scrolling the input horizontally on a single screen line when it
becomes longer than the screen width rather than wrapping to a new line.
.TP
.B input\-meta (Off)
If set to \fBOn\fP, readline will enable eight-bit input (that is,
it will not strip the high bit from the characters it reads),
regardless of what the terminal claims it can support. The name
.B meta\-flag
is a synonym for this variable.
.TP
.B isearch\-terminators (``C\-[C\-J'')
The string of characters that should terminate an incremental
search without subsequently executing the character as a command.
If this variable has not been given a value, the characters
\fIESC\fP and \fIC\-J\fP will terminate an incremental search.
.TP
.B keymap (emacs)
Set the current readline keymap. The set of valid keymap names is
\fIemacs, emacs\-standard, emacs\-meta, emacs\-ctlx, vi,
vi\-command\fP, and
.IR vi\-insert .
\fIvi\fP is equivalent to \fIvi\-command\fP; \fIemacs\fP is
equivalent to \fIemacs\-standard\fP. The default value is
.IR emacs ;
the value of
.B editing\-mode
also affects the default keymap.
.TP
.B mark\-directories (On)
If set to \fBOn\fP, completed directory names have a slash
appended.
.TP
.B mark\-modified\-lines (Off)
If set to \fBOn\fP, history lines that have been modified are displayed
with a preceding asterisk (\fB*\fP).
.TP
.B mark\-symlinked\-directories (Off)
If set to \fBOn\fP, completed names which are symbolic links to directories
have a slash appended (subject to the value of
\fBmark\-directories\fP).
.TP
.B match\-hidden\-files (On)
This variable, when set to \fBOn\fP, causes readline to match files whose
names begin with a `.' (hidden files) when performing filename
completion, unless the leading `.' is

```
supplied by the user in the filename to be completed.
.TP
.B output\-meta (Off)
If set to \fBOn\fP, readline will display characters with the
eighth bit set directly rather than as a meta-prefixed escape
sequence.
.TP
.B page\-completions (On)
If set to \fBOn\fP, readline uses an internal \fImore\fP-like pager
to display a screenful of possible completions at a time.
.TP
.B print\-completions\-horizontally (Off)
If set to \fBOn\fP, readline will display completions with matches
sorted horizontally in alphabetical order, rather than down the screen.
.TP
.B show\-all\-if\-ambiguous (Off)
This alters the default behavior of the completion functions.  If
set to
.BR on ,
words which have more than one possible completion cause the
matches to be listed immediately instead of ringing the bell.
.TP
.B show\-all\-if\-unmodified (Off)
This alters the default behavior of the completion functions in
a fashion similar to \fBshow\-all\-if\-ambiguous\fP.
If set to
.BR on ,
words which have more than one possible completion without any
possible partial completion (the possible completions don't share
a common prefix) cause the matches to be listed immediately instead
of ringing the bell.
.TP
.B visible\-stats (Off)
If set to \fBOn\fP, a character denoting a file's type as reported
by \fIstat\fP(2) is appended to the filename when listing possible
completions.
.PD
.SS "Readline Conditional Constructs"
.PP
Readline implements a facility similar in spirit to the conditional
compilation features of the C preprocessor which allows key
bindings and variable settings to be performed as the result
of tests.  There are four parser directives used.
.IP \fB$if\fP
The
.B $if
construct allows bindings to be made based on the
editing mode, the terminal being used, or the application using
readline.  The text of the test extends to the end of the line;
no characters are required to isolate it.
.RS
.IP \fBmode\fP
The \fBmode=\fP form of the \fB$if\fP directive is used to test
whether readline is in emacs or vi mode.
This may be used in conjunction
with the \fBset keymap\fP command, for instance, to set bindings in
the \fIemacs\-standard\fP and \fIemacs\-ctlx\fP keymaps only if
readline is starting out in emacs mode.
.IP \fBterm\fP
The \fBterm=\fP form may be used to include terminal-specific
key bindings, perhaps to bind the key sequences output by the
terminal's function keys.  The word on the right side of the
.B =
is tested against the both full name of the terminal and the portion
of the terminal name before the first \fB\-\fP.  This allows
.I sun
to match both
.I sun
and
.IR sun\-cmd ,
for instance.
.IP \fBapplication\fP
The \fBapplication\fP construct is used to include
application-specific settings.  Each program using the readline
library sets the \fIapplication name\fP, and an initialization
file can test for a particular value.
This could be used to bind key sequences to functions useful for
a specific program.  For instance, the following command adds a
key sequence that quotes the current or previous word in Bash:
.sp 1
.RS
.nf
\fB$if\fP Bash
# Quote the current or previous word
"\eC\-xq": "\eeb\e"\eef\e""
\fB$endif\fP
.fi
.RE
.RE
.IP \fB$endif\fP
This command, as seen in the previous example, terminates an
\fB$if\fP command.
.IP \fB$else\fP
Commands in this branch of the \fB$if\fP directive are executed if
the test fails.
.IP \fB$include\fP
This directive takes a single filename as an argument and reads commands
and bindings from that file.  For example, the following directive
would read \fI/etc/inputrc\fP:
.sp 1
.RS
.nf
\fB$include\fP \^ \fI/etc/inputrc\fP
.fi
.RE
.SS Searching
.PP
Readline provides commands for searching through the command history
(see
.SM
.B HISTORY
below) for lines containing a specified string.
There are two search modes:
.I incremental
and
.IR non-incremental .
.PP
Incremental searches begin before the user has finished typing the
search string.
As each character of the search string is typed, readline displays
the next entry from the history matching the string typed so far.
An incremental search requires only as many characters as needed to
```

find the desired history entry.
The characters present in the value of the \fBisearch-terminators\fP
variable are used to terminate an incremental search.
If that variable has not been assigned a value the Escape and
Control-J characters will terminate an incremental search.
Control-G will abort an incremental search and restore the original
line.
When the search is terminated, the history entry containing the
search string becomes the current line.
.PP
To find other matching entries in the history list, type Control-S or
Control-R as appropriate.
This will search backward or forward in the history for the next
entry matching the search string typed so far.
Any other key sequence bound to a readline command will terminate
the search and execute that command.
For instance, a \fInewline\fP will terminate the search and accept
the line, thereby executing the command from the history list.
.PP
Readline remembers the last incremental search string. If two
Control-Rs are typed without any intervening characters defining a
new search string, any remembered search string is used.
.PP
Non-incremental searches read the entire search string before starting
to search for matching history lines. The search string may be
typed by the user or be part of the contents of the current line.
.SS "Readline Command Names"
.PP
The following is a list of the names of the commands and the default
key sequences to which they are bound.
Command names without an accompanying key sequence are unbound by default.
In the following descriptions, \fIpoint\fP refers to the current cursor
position, and \fImark\fP refers to a cursor position saved by the
\fBset\-mark\fP command.
The text between the point and mark is referred to as the \fIregion\fP.
.SS Commands for Moving
.PP
.PD 0
.TP
.B beginning\-of\-line (C\-a)
Move to the start of the current line.
.TP
.B end\-of\-line (C\-e)
Move to the end of the line.
.TP
.B forward\-char (C\-f)
Move forward a character.
.TP
.B backward\-char (C\-b)
Move back a character.
.TP
.B forward\-word (M\-f)
Move forward to the end of the next word. Words are composed of
alphanumeric characters (letters and digits).
.TP
.B backward\-word (M\-b)
Move back to the start of the current or previous word. Words are
composed of alphanumeric characters (letters and digits).
.TP
.B clear\-screen (C\-l)
Clear the screen leaving the current line at the top of the screen.
With an argument, refresh the current line without clearing the
screen.
.TP
.B redraw\-current\-line
Refresh the current line.
.PD
.SS Commands for Manipulating the History
.PP
.PD 0
.TP
.B accept\-line (Newline, Return)
Accept the line regardless of where the cursor is. If this line is
non-empty, add it to the history list according to the state of the
.SM
.B HISTCONTROL
variable. If the line is a modified history
line, then restore the history line to its original state.
.TP
.B previous\-history (C\-p)
Fetch the previous command from the history list, moving back in
the list.
.TP
.B next\-history (C\-n)
Fetch the next command from the history list, moving forward in the
list.
.TP
.B beginning\-of\-history (M\-<)
Move to the first line in the history.
.TP
.B end\-of\-history (M\->)
Move to the end of the input history, i.e., the line currently being
entered.
.TP
.B reverse\-search\-history (C\-r)
Search backward starting at the current line and moving `up' through
the history as necessary. This is an incremental search.
.TP
.B forward\-search\-history (C\-s)
Search forward starting at the current line and moving `down' through
the history as necessary. This is an incremental search.
.TP
.B non\-incremental\-reverse\-search\-history (M\-p)
Search backward through the history starting at the current line
using a non-incremental search for a string supplied by the user.
.TP
.B non\-incremental\-forward\-search\-history (M\-n)
Search forward through the history using a non-incremental search for
a string supplied by the user.
.TP
.B history\-search\-forward
Search forward through the history for the string of characters
between the start of the current line and the point.
This is a non-incremental search.
.TP
.B history\-search\-backward
Search backward through the history for the string of characters
between the start of the current line and the point.
This is a non-incremental search.
.TP
.B yank\-nth\-arg (M\-C\-y)
Insert the first argument to the previous command (usually

the second word on the previous line) at point.
With an argument
.IR n ,
insert the \fIn\fPth word from the previous command (the words
in the previous command begin with word 0). A negative argument
inserts the \fIn\fPth word from the end of the previous command.
Once the argument \fIn\fP is computed, the argument is extracted
as if the "!\fIn\fP" history expansion had been specified.
.TP
.B
yank\-last\-arg (M\-.\^, M\-_\^)
Insert the last argument to the previous command (the last word of
the previous history entry). With an argument,
behave exactly like \fByank\-nth\-arg\fP.
Successive calls to \fByank\-last\-arg\fP move back through the history
list, inserting the last argument of each line in turn.
The history expansion facilities are used to extract the last argument,
as if the "!$" history expansion had been specified.
.TP
.B shell\-expand\-line (M\-C\-e)
Expand the line as the shell does. This
performs alias and history expansion as well as all of the shell
word expansions. See
.SM
.B HISTORY EXPANSION
below for a description of history expansion.
.TP
.B history\-expand\-line (M\-^)
Perform history expansion on the current line.
See
.SM
.B HISTORY EXPANSION
below for a description of history expansion.
.TP
.B magic\-space
Perform history expansion on the current line and insert a space.
See
.SM
.B HISTORY EXPANSION
below for a description of history expansion.
.TP
.B alias\-expand\-line
Perform alias expansion on the current line.
See
.SM
.B ALIASES
above for a description of alias expansion.
.TP
.B history\-and\-alias\-expand\-line
Perform history and alias expansion on the current line.
.TP
.B insert\-last\-argument (M\-.\^, M\-_\^)
A synonym for \fByank\-last\-arg\fP.
.TP
.B operate\-and\-get\-next (C\-o)
Accept the current line for execution and fetch the next line
relative to the current line from the history for editing. Any
argument is ignored.
.TP
.B edit\-and\-execute\-command (C\-xC\-e)
Invoke an editor on the current command line, and execute the result as shell
commands.
\fBBash\fP attempts to invoke
.SM
.BR $FCEDIT ,
.SM
.BR $EDITOR ,
and \fIemacs\fP as the editor, in that order.
.PD
.SS Commands for Changing Text
.PP
.PD 0
.TP
.B delete\-char (C\-d)
Delete the character at point. If point is at the
beginning of the line, there are no characters in the line, and
the last character typed was not bound to \fBdelete\-char\fP,
then return
.SM
.BR EOF .
.TP
.B backward\-delete\-char (Rubout)
Delete the character behind the cursor. When given a numeric argument,
save the deleted text on the kill ring.
.TP
.B forward\-backward\-delete\-char
Delete the character under the cursor, unless the cursor is at the
end of the line, in which case the character behind the cursor is
deleted.
.TP
.B quoted\-insert (C\-q, C\-v)
Add the next character typed to the line verbatim. This is
how to insert characters like \fBC\-q\fP, for example.
.TP
.B tab\-insert (C\-v TAB)
Insert a tab character.
.TP
.B self\-insert (a,\ b,\ A,\ 1,\ !,\ ...)
Insert the character typed.
.TP
.B transpose\-chars (C\-t)
Drag the character before point forward over the character at point,
moving point forward as well.
If point is at the end of the line, then this transposes
the two characters before point.
Negative arguments have no effect.
.TP
.B transpose\-words (M\-t)
Drag the word before point past the word after point,
moving point over that word as well.
If point is at the end of the line, this transposes
the last two words on the line.
.TP
.B upcase\-word (M\-u)
Uppercase the current (or following) word. With a negative argument,
uppercase the previous word, but do not move point.
.TP
.B downcase\-word (M\-l)
Lowercase the current (or following) word. With a negative argument,
lowercase the previous word, but do not move point.
.TP
.B capitalize\-word (M\-c)

Capitalize the current (or following) word. With a negative argument,
capitalize the previous word, but do not move point.
.TP
.B overwrite\-mode
Toggle overwrite mode. With an explicit positive numeric argument,
switches to overwrite mode. With an explicit non-positive numeric
argument, switches to insert mode. This command affects only
\fBemacs\fP mode; \fBvi\fP mode does overwrite differently.
Each call to \fIreadline()\fP starts in insert mode.
In overwrite mode, characters bound to \fBself\-insert\fP replace
the text at point rather than pushing the text to the right.
Characters bound to \fBbackward\-delete\-char\fP replace the character
before point with a space. By default, this command is unbound.
.PD
.SS Killing and Yanking
.PP
.PD 0
.TP
.B kill\-line (C\-k)
Kill the text from point to the end of the line.
.TP
.B backward\-kill\-line (C\-x Rubout)
Kill backward to the beginning of the line.
.TP
.B unix\-line\-discard (C\-u)
Kill backward from point to the beginning of the line.
The killed text is saved on the kill-ring.
.\" There is no real difference between this and backward-kill-line
.TP
.B kill\-whole\-line
Kill all characters on the current line, no matter where point is.
.TP
.B kill\-word (M\-d)
Kill from point to the end of the current word, or if between
words, to the end of the next word.
Word boundaries are the same as those used by \fBforward\-word\fP.
.TP
.B backward\-kill\-word (M\-Rubout)
Kill the word behind point.
Word boundaries are the same as those used by \fBbackward\-word\fP.
.TP
.B unix\-word\-rubout (C\-w)
Kill the word behind point, using white space as a word boundary.
The killed text is saved on the kill-ring.
.TP
.B unix\-filename\-rubout
Kill the word behind point, using white space and the slash character
as the word boundaries.
The killed text is saved on the kill-ring.
.TP
.B delete\-horizontal\-space (M\-\e)
Delete all spaces and tabs around point.
.TP
.B kill\-region
Kill the text in the current region.
.TP
.B copy\-region\-as\-kill
Copy the text in the region to the kill buffer.
.TP
.B copy\-backward\-word
Copy the word before point to the kill buffer.
The word boundaries are the same as \fBbackward\-word\fP.
.TP
.B copy\-forward\-word
Copy the word following point to the kill buffer.
The word boundaries are the same as \fBforward\-word\fP.
.TP
.B yank (C\-y)
Yank the top of the kill ring into the buffer at point.
.TP
.B yank\-pop (M\-y)
Rotate the kill ring, and yank the new top. Only works following
.B yank
or
.BR yank\-pop .
.PD
.SS Numeric Arguments
.PP
.PD 0
.TP
.B digit\-argument (M\-0, M\-1, ..., M\-\-)
Add this digit to the argument already accumulating, or start a new
argument. M\-\- starts a negative argument.
.TP
.B universal\-argument
This is another way to specify an argument.
If this command is followed by one or more digits, optionally with a
leading minus sign, those digits define the argument.
If the command is followed by digits, executing
.B universal\-argument
again ends the numeric argument, but is otherwise ignored.
As a special case, if this command is immediately followed by a
character that is neither a digit or minus sign, the argument count
for the next command is multiplied by four.
The argument count is initially one, so executing this function the
first time makes the argument count four, a second time makes the
argument count sixteen, and so on.
.PD
.SS Completing
.PP
.PD 0
.TP
.B complete (TAB)
Attempt to perform completion on the text before point.
.B Bash
attempts completion treating the text as a variable (if the
text begins with \fB$\fP), username (if the text begins with
\fB~\fP), hostname (if the text begins with \fB@\fP), or
command (including aliases and functions) in turn. If none
of these produces a match, filename completion is attempted.
.TP
.B possible\-completions (M\-?)
List the possible completions of the text before point.
.TP
.B insert\-completions (M\-*)
Insert all completions of the text before point
that would have been generated by
\fBpossible\-completions\fP.
.TP
.B menu\-complete
Similar to \fBcomplete\fP, but replaces the word to be completed
with a single match from the list of possible completions.

Repeated execution of \fBmenu\-complete\fP steps through the list
of possible completions, inserting each match in turn.
At the end of the list of completions, the bell is rung
(subject to the setting of \fBbell\-style\fP)
and the original text is restored.
An argument of \fIn\fP moves \fIn\fP positions forward in the list
of matches; a negative argument may be used to move backward
through the list.
This command is intended to be bound to \fBTAB\fP, but is unbound
by default.
.TP
.B delete\-char\-or\-list
Deletes the character under the cursor if not at the beginning or
end of the line (like \fBdelete\-char\fP).
If at the end of the line, behaves identically to
\fBpossible\-completions\fP.
This command is unbound by default.
.TP
.B complete\-filename (M\-/)
Attempt filename completion on the text before point.
.TP
.B possible\-filename\-completions (C\-x /)
List the possible completions of the text before point,
treating it as a filename.
.TP
.B complete\-username (M\-~)
Attempt completion on the text before point, treating
it as a username.
.TP
.B possible\-username\-completions (C\-x ~)
List the possible completions of the text before point,
treating it as a username.
.TP
.B complete\-variable (M\-$)
Attempt completion on the text before point, treating
it as a shell variable.
.TP
.B possible\-variable\-completions (C\-x $)
List the possible completions of the text before point,
treating it as a shell variable.
.TP
.B complete\-hostname (M\-@)
Attempt completion on the text before point, treating
it as a hostname.
.TP
.B possible\-hostname\-completions (C\-x @)
List the possible completions of the text before point,
treating it as a hostname.
.TP
.B complete\-command (M\-!)
Attempt completion on the text before point, treating
it as a command name. Command completion attempts to
match the text against aliases, reserved words, shell
functions, shell builtins, and finally executable filenames,
in that order.
.TP
.B possible\-command\-completions (C\-x !)
List the possible completions of the text before point,
treating it as a command name.
.TP
.B dynamic\-complete\-history (M\-TAB)
Attempt completion on the text before point, comparing
the text against lines from the history list for possible
completion matches.
.TP
.B complete\-into\-braces (M\-{)
Perform filename completion and insert the list of possible completions
enclosed within braces so the list is available to the shell (see
.B Brace Expansion
above).
.PD
.SS Keyboard Macros
.PP
.PD 0
.TP
.B start\-kbd\-macro (C\-x (\^)
Begin saving the characters typed into the current keyboard macro.
.TP
.B end\-kbd\-macro (C\-x)\^)
Stop saving the characters typed into the current keyboard macro
and store the definition.
.TP
.B call\-last\-kbd\-macro (C\-x e)
Re-execute the last keyboard macro defined, by making the characters
in the macro appear as if typed at the keyboard.
.PD
.SS Miscellaneous
.PP
.PD 0
.TP
.B re\-read\-init\-file (C\-x C\-r)
Read in the contents of the \fIinputrc\fP file, and incorporate
any bindings or variable assignments found there.
.TP
.B abort (C\-g)
Abort the current editing command and
ring the terminal's bell (subject to the setting of
.BR bell\-style).
.TP
.B do\-uppercase\-version (M\-a, M\-b, M\-\fIx\fP, ...)
If the metafied character \fIx\fP is lowercase, run the command
that is bound to the corresponding uppercase character.
.TP
.B prefix\-meta (ESC)
Metafy the next character typed.
.SM
.B ESC
.B f
is equivalent to
.BR Meta\-f .
.TP
.B undo (C\-_, C\-x C\-u)
Incremental undo, separately remembered for each line.
.TP
.B revert\-line (M\-r)
Undo all changes made to this line. This is like executing the
.B undo
command enough times to return the line to its initial state.
.TP
.B tilde\-expand (M\-&)
Perform tilde expansion on the current word.
.TP

.B set\-mark (C\-@, M\-<space>)
Set the mark to the point. If a
numeric argument is supplied, the mark is set to that position.
.TP
.B exchange\-point\-and\-mark (C\-x C\-x)
Swap the point with the mark. The current cursor position is set to
the saved position, and the old cursor position is saved as the mark.
.TP
.B character\-search (C\-])
A character is read and point is moved to the next occurrence of that
character. A negative count searches for previous occurrences.
.TP
.B character\-search\-backward (M\-C\-])
A character is read and point is moved to the previous occurrence of that
character. A negative count searches for subsequent occurrences.
.TP
.B insert\-comment (M\-#)
Without a numeric argument, the value of the readline
.B comment\-begin
variable is inserted at the beginning of the current line.
If a numeric argument is supplied, this command acts as a toggle: if
the characters at the beginning of the line do not match the value
of \fBcomment\-begin\fP, the value is inserted, otherwise
the characters in \fBcomment-begin\fP are deleted from the beginning of
the line.
In either case, the line is accepted as if a newline had been typed.
The default value of
\fBcomment\-begin\fP causes this command to make the current line
a shell comment.
If a numeric argument causes the comment character to be removed, the line
will be executed by the shell.
.TP
.B glob\-complete\-word (M\-g)
The word before point is treated as a pattern for pathname expansion,
with an asterisk implicitly appended. This pattern is used to
generate a list of matching file names for possible completions.
.TP
.B glob\-expand\-word (C\-x *)
The word before point is treated as a pattern for pathname expansion,
and the list of matching file names is inserted, replacing the word.
If a numeric argument is supplied, an asterisk is appended before
pathname expansion.
.TP
.B glob\-list\-expansions (C\-x g)
The list of expansions that would have been generated by
.B glob\-expand\-word
is displayed, and the line is redrawn.
If a numeric argument is supplied, an asterisk is appended before
pathname expansion.
.TP
.B dump\-functions
Print all of the functions and their key bindings to the
readline output stream. If a numeric argument is supplied,
the output is formatted in such a way that it can be made part
of an \fIinputrc\fP file.
.TP
.B dump\-variables
Print all of the settable readline variables and their values to the
readline output stream. If a numeric argument is supplied,
the output is formatted in such a way that it can be made part
of an \fIinputrc\fP file.
.TP
.B dump\-macros
Print all of the readline key sequences bound to macros and the
strings they output. If a numeric argument is supplied,
the output is formatted in such a way that it can be made part
of an \fIinputrc\fP file.
.TP
.B display\-shell\-version (C\-x C\-v)
Display version information about the current instance of
.BR bash .
.PD
.SS Programmable Completion
.PP
When word completion is attempted for an argument to a command for
which a completion specification (a \fIcompspec\fP) has been defined
using the \fBcomplete\fP builtin (see
.SM
.B "SHELL BUILTIN COMMANDS"
below), the programmable completion facilities are invoked.
.PP
First, the command name is identified.
If a compspec has been defined for that command, the
compspec is used to generate the list of possible completions for the word.
If the command word is a full pathname, a compspec for the full
pathname is searched for first.
If no compspec is found for the full pathname, an attempt is made to
find a compspec for the portion following the final slash.
.PP
Once a compspec has been found, it is used to generate the list of
matching words.
If a compspec is not found, the default \fBbash\fP completion as
described above under \fBCompleting\fP is performed.
.PP
First, the actions specified by the compspec are used.
Only matches which are prefixed by the word being completed are
returned.
When the
.B \-f
or
.B \-d
option is used for filename or directory name completion, the shell
variable
.SM
.B FIGNORE
is used to filter the matches.
.PP
Any completions specified by a filename expansion pattern to the
\fB\-G\fP option are generated next.
The words generated by the pattern need not match the word
being completed.
The
.SM
.B GLOBIGNORE
shell variable is not used to filter the matches, but the
.SM
.B FIGNORE
variable is used.
.PP
Next, the string specified as the argument to the \fB\-W\fP option
is considered.
The string is first split using the characters in the

.SM
.B IFS
special variable as delimiters.
Shell quoting is honored.
Each word is then expanded using
brace expansion, tilde expansion, parameter and variable expansion,
command substitution, and arithmetic expansion,
as described above under
.SM
.BR EXPANSION .
The results are split using the rules described above under
\fBWord Splitting\fP.
The results of the expansion are prefix-matched against the word being
completed, and the matching words become the possible completions.
.PP
After these matches have been generated, any shell function or command
specified with the \fB\-F\fP and \fB\-C\fP options is invoked.
When the command or function is invoked, the
.SM
.B COMP_LINE
and
.SM
.B COMP_POINT
variables are assigned values as described above under
\fBShell Variables\fP.
If a shell function is being invoked, the
.SM
.B COMP_WORDS
and
.SM
.B COMP_CWORD
variables are also set.
When the function or command is invoked, the first argument is the
name of the command whose arguments are being completed, the
second argument is the word being completed, and the third argument
is the word preceding the word being completed on the current command line.
No filtering of the generated completions against the word being completed
is performed; the function or command has complete freedom in generating
the matches.
.PP
Any function specified with \fB\-F\fP is invoked first.
The function may use any of the shell facilities, including the
\fBcompgen\fP builtin described below, to generate the matches.
It must put the possible completions in the
.SM
.B COMPREPLY
array variable.
.PP
Next, any command specified with the \fB\-C\fP option is invoked
in an environment equivalent to command substitution.
It should print a list of completions, one per line, to the
standard output.
Backslash may be used to escape a newline, if necessary.
.PP
After all of the possible completions are generated, any filter
specified with the \fB\-X\fP option is applied to the list.
The filter is a pattern as used for pathname expansion; a \fB&\fP
in the pattern is replaced with the text of the word being completed.
A literal \fB&\fP may be escaped with a backslash; the backslash
is removed before attempting a match.
Any completion that matches the pattern will be removed from the list.
A leading \fB!\fP negates the pattern; in this case any completion
not matching the pattern will be removed.
.PP
Finally, any prefix and suffix specified with the \fB\-P\fP and \fB\-S\fP
options are added to each member of the completion list, and the result is
returned to the readline completion code as the list of possible
completions.
.PP
If the previously-applied actions do not generate any matches, and the
\fB\-o dirnames\fP option was supplied to \fBcomplete\fP when the
compspec was defined, directory name completion is attempted.
.PP
If the \fB\-o plusdirs\fP option was supplied to \fBcomplete\fP when the
compspec was defined, directory name completion is attempted and any
matches are added to the results of the other actions.
.PP
By default, if a compspec is found, whatever it generates is returned
to the completion code as the full set of possible completions.
The default \fBbash\fP completions are not attempted, and the readline
default of filename completion is disabled.
If the \fB\-o bashdefault\fP option was supplied to \fBcomplete\fP when
the compspec was defined, the \fBbash\fP default completions are attempted
if the compspec generates no matches.
If the \fB\-o default\fP option was supplied to \fBcomplete\fP when the
compspec was defined, readline's default completion will be performed
if the compspec (and, if attempted, the default \fBbash\fP completions)
generate no matches.
.PP
When a compspec indicates that directory name completion is desired,
the programmable completion functions force readline to append a slash
to completed names which are symbolic links to directories, subject to
the value of the \fBmark\-directories\fP readline variable, regardless
of the setting of the \fBmark-symlinked\-directories\fP readline variable.
.SH HISTORY
When the
.B \-o history
option to the
.B set
builtin is enabled, the shell provides access to the
\fIcommand history\fP,
the list of commands previously typed.
The value of the \fBHISTSIZE\fP variable is used as the
number of commands to save in a history list.
The text of the last
.SM
.B HISTSIZE
commands (default 500) is saved. The shell
stores each command in the history list prior to parameter and
variable expansion (see
.SM
.B EXPANSION
above) but after history expansion is performed, subject to the
values of the shell variables
.SM
.B HISTIGNORE
and
.SM
.BR HISTCONTROL .
.PP
On startup, the history is initialized from the file named by
the variable

.SM
.B HISTFILE
(default \fI~/.bash_history\fP).
The file named by the value of
.SM
.B HISTFILE
is truncated, if necessary, to contain no more than
the number of lines specified by the value of
.SM
.BR HISTFILESIZE .
When an interactive shell exits, the last
.SM
.B $HISTSIZE
lines are copied from the history list to
.SM
.BR $HISTFILE .
If the
.B histappend
shell option is enabled
(see the description of
.B shopt
under
.SM
.B "SHELL BUILTIN COMMANDS"
below), the lines are appended to the history file,
otherwise the history file is overwritten.
If
.SM
.B HISTFILE
is unset, or if the history file is unwritable, the history is
not saved. After saving the history, the history file is truncated
to contain no more than
.SM
.B HISTFILESIZE
lines. If
.SM
.B HISTFILESIZE
is not set, no truncation is performed.
.PP
The builtin command
.B fc
(see
.SM
.B SHELL BUILTIN COMMANDS
below) may be used to list or edit and re-execute a portion of
the history list.
The
.B history
builtin may be used to display or modify the history list and
manipulate the history file.
When using command-line editing, search commands
are available in each editing mode that provide access to the
history list.
.PP
The shell allows control over which commands are saved on the history
list. The
.SM
.B HISTCONTROL
and
.SM
.B HISTIGNORE
variables may be set to cause the shell to save only a subset of the
commands entered.
The
.B cmdhist
shell option, if enabled, causes the shell to attempt to save each
line of a multi-line command in the same history entry, adding
semicolons where necessary to preserve syntactic correctness.
The
.B lithist
shell option causes the shell to save the command with embedded newlines
instead of semicolons. See the description of the
.B shopt
builtin below under
.SM
.B "SHELL BUILTIN COMMANDS"
for information on setting and unsetting shell options.
.SH "HISTORY EXPANSION"
.PP
The shell supports a history expansion feature that
is similar to the history expansion in
.BR csh.
This section describes what syntax features are available. This
feature is enabled by default for interactive shells, and can be
disabled using the
.B \+H
option to the
.B set
builtin command (see
.SM
.B SHELL BUILTIN COMMANDS
below). Non-interactive shells do not perform history expansion
by default.
.PP
History expansions introduce words from the history list into
the input stream, making it easy to repeat commands, insert the
arguments to a previous command into the current input line, or
fix errors in previous commands quickly.
.PP
History expansion is performed immediately after a complete line
is read, before the shell breaks it into words.
It takes place in two parts.
The first is to determine which line from the history list
to use during substitution.
The second is to select portions of that line for inclusion into
the current one.
The line selected from the history is the \fIevent\fP,
and the portions of that line that are acted upon are \fIwords\fP.
Various \fImodifiers\fP are available to manipulate the selected words.
The line is broken into words in the same fashion as when reading input,
so that several \fImetacharacter\fP-separated words surrounded by
quotes are considered one word.
History expansions are introduced by the appearance of the
history expansion character, which is \^\fB!\fP\^ by default.
Only backslash (\^\fB\e\fP\^) and single quotes can quote
the history expansion character.
.PP
Several characters inhibit history expansion if found immediately
following the history expansion character, even if it is unquoted:
space, tab, newline, carriage return, and \fB=\fP.
If the \fBextglob\fP shell option is enabled, \fB(\fP will also
inhibit expansion.

```
.PP
Several shell options settable with the
.B shopt
builtin may be used to tailor the behavior of history expansion.
If the
.B histverify
shell option is enabled (see the description of the
.B shopt
builtin), and
.B readline
is being used, history substitutions are not immediately passed to
the shell parser.
Instead, the expanded line is reloaded into the
.B readline
editing buffer for further modification.
If
.B readline
is being used, and the
.B histreedit
shell option is enabled, a failed history substitution will be reloaded
into the
.B readline
editing buffer for correction.
The
.B \-p
option to the
.B history
builtin command may be used to see what a history expansion will
do before using it.
The
.B \-s
option to the
.B history
builtin may be used to add commands to the end of the history list
without actually executing them, so that they are available for
subsequent recall.
.PP
The shell allows control of the various characters used by the
history expansion mechanism (see the description of
.B histchars
above under
.BR "Shell Variables" ).
.SS Event Designators
.PP
An event designator is a reference to a command line entry in the
history list.
.PP
.PD 0
.TP
.B !
Start a history substitution, except when followed by a
.BR blank ,
newline, carriage return, =
or ( (when the \fBextglob\fP shell option is enabled using
the \fBshopt\fP builtin).
.TP
.B !\fIn\fR
Refer to command line
.IR n .
.TP
.B !\-\fIn\fR
Refer to the current command line minus
.IR n .
.TP
.B !!
Refer to the previous command.  This is a synonym for `!\-1'.
.TP
.B !\fIstring\fR
Refer to the most recent command starting with
.IR string .
.TP
.B !?\fIstring\fR\fB[?]\fR
Refer to the most recent command containing
.IR string .
The trailing \fB?\fP may be omitted if
.I string
is followed immediately by a newline.
.TP
.B \d\s+2^\s-2\u\fIstring1\fP\d\s+2^\s-2\u\fIstring2\fP\d\s+2^\s-2\u
Quick substitution.  Repeat the last command, replacing
.I string1
with
.IR string2 .
Equivalent to
``!!:s/\fIstring1\fP/\fIstring2\fP/''
(see \fBModifiers\fP below).
.TP
.B !#
The entire command line typed so far.
.PD
.SS Word Designators
.PP
Word designators are used to select desired words from the event.
A
.B :
separates the event specification from the word designator.
It may be omitted if the word designator begins with a
.BR ^ ,
.BR $ ,
.BR * ,
.BR \- ,
or
.BR % .
Words are numbered from the beginning of the line,
with the first word being denoted by 0 (zero).
Words are inserted into the current line separated by single spaces.
.PP
.PD 0
.TP
.B 0 (zero)
The zeroth word.  For the shell, this is the command
word.
.TP
.I n
The \fIn\fRth word.
.TP
.B ^
The first argument.  That is, word 1.
.TP
.B $
The last argument.
.TP
```

.B %
The word matched by the most recent `?\fIstring\fR?' search.
.TP
.I x\fB\-\fPy
A range of words; `\-\fIy\fR' abbreviates `0\-\fIy\fR'.
.TP
.B *
All of the words but the zeroth. This is a synonym
for `\fI1\-$\fP'. It is not an error to use
.B *
if there is just one
word in the event; the empty string is returned in that case.
.TP
.B x*
Abbreviates \fIx\-$\fP.
.TP
.B x\-
Abbreviates \fIx\-$\fP like \fBx*\fP, but omits the last word.
.PD
.PP
If a word designator is supplied without an event specification, the
previous command is used as the event.
.SS Modifiers
.PP
After the optional word designator, there may appear a sequence of
one or more of the following modifiers, each preceded by a `:'.
.PP
.PD 0
.PP
.TP
.B h
Remove a trailing file name component, leaving only the head.
.TP
.B t
Remove all leading file name components, leaving the tail.
.TP
.B r
Remove a trailing suffix of the form \fI.xxx\fP, leaving the
basename.
.TP
.B e
Remove all but the trailing suffix.
.TP
.B p
Print the new command but do not execute it.
.TP
.B q
Quote the substituted words, escaping further substitutions.
.TP
.B x
Quote the substituted words as with
.BR q ,
but break into words at
.B blanks
and newlines.
.TP
.B s/\fIold\fP/\fInew\fP/
Substitute
.I new
for the first occurrence of
.I old
in the event line. Any delimiter can be used in place of /. The
final delimiter is optional if it is the last character of the
event line. The delimiter may be quoted in
.I old
and
.I new
with a single backslash. If & appears in
.IR new ,
it is replaced by
.IR old .
A single backslash will quote the &. If
.I old
is null, it is set to the last
.I old
substituted, or, if no previous history substitutions took place,
the last
.I string
in a
.B !?\fIstring\fR\fB[?]\fR
search.
.TP
.B &
Repeat the previous substitution.
.TP
.B g
Cause changes to be applied over the entire event line. This is
used in conjunction with `\fB:s\fP' (e.g., `\fB:gs/\fIold\fP/\fInew\fP/\fR')
or `\fB:&\fP'. If used with
`\fB:s\fP', any delimiter can be used
in place of /, and the final delimiter is optional
if it is the last character of the event line.
An \fBa\fP may be used as a synonym for \fBg\fP.
.TP
.B G
Apply the following `\fBs\fP' modifier once to each word in the event line.
.PD
.SH "SHELL BUILTIN COMMANDS"
.\" start of bash_builtins
.zZ
.PP
Unless otherwise noted, each builtin command documented in this
section as accepting options preceded by
.B \-
accepts
.B \-\-
to signify the end of the options.
For example, the \fB:\fP, \fBtrue\fP, \fBfalse\fP, and \fBtest\fP builtins
do not accept options.
.sp .5
.PD 0
.TP
\fB:\fP [\fIarguments\fP]
.PD
No effect; the command does nothing beyond expanding
.I arguments
and performing any specified
redirections. A zero exit code is returned.
.TP
\fB .\|.\| \fP \fIfilename\fP [\fIarguments\fP]
.PD 0
.TP

```
\fBsource\fP \fIfilename\fP [\fIarguments\fP]
.PD
Read and execute commands from
.I filename
in the current
shell environment and return the exit status of the last command
executed from
.IR filename .
If
.I filename
does not contain a slash, file names in
.SM
.B PATH
are used to find the directory containing
.IR filename .
The file searched for in
.SM
.B PATH
need not be executable.
When \fBbash\fP is not in \fIposix mode\fP, the current directory is
searched if no file is found in
.SM
.BR PATH .
If the
.B sourcepath
option to the
.B shopt
builtin command is turned off, the
.SM
.B PATH
is not searched.
If any \fIarguments\fP are supplied, they become the positional
parameters when \fIfilename\fP is executed.  Otherwise the positional
parameters are unchanged.
The return status is the status of the last command exited within
the script (0 if no commands are executed), and false if
.I filename
is not found or cannot be read.
.TP
\fBalias\fP [\fB\-p\fP] [\fIname\fP[=\fIvalue\fP] ...]
\fBAlias\fP with no arguments or with the
.B \-p
option prints the list of aliases in the form
\fBalias\fP \fIname\fP=\fIvalue\fP on standard output.
When arguments are supplied, an alias is defined for
each \fIname\fP whose \fIvalue\fP is given.
A trailing space in  \fIvalue\fP causes the next word to be
checked for alias substitution when the alias is expanded.
For each \fIname\fP in the argument list for which no \fIvalue\fP
is supplied, the name and value of the alias is printed.
\fBAlias\fP returns true unless a \fIname\fP is given for which
no alias has been defined.
.TP
\fBbg\fP [\fIjobspec\fP ...]
Resume each suspended job \fIjobspec\fP in the background, as if it
had been started with
.BR & .
If \fIjobspec\fP is not present, the shell's notion of the
\fIcurrent job\fP is used.
.B bg
.I jobspec
returns 0 unless run when job control is disabled or, when run with
job control enabled, any specified \fIjobspec\fP was not found
or was started without job control.
.TP
\fBbind\fP [\fB\-m\fP \fIkeymap\fP] [\fB\-lpsvPSV\fP]
.PD 0
.TP
\fBbind\fP [\fB\-m\fP \fIkeymap\fP] [\fB\-q\fP \fIfunction\fP] [\fB\-u\fP \fIfunction\fP] [\fB\-r\fP \fIkeyseq\fP]
.TP
\fBbind\fP [\fB\-m\fP \fIkeymap\fP] \fB\-f\fP \fIfilename\fP
.TP
\fBbind\fP [\fB\-m\fP \fIkeymap\fP] \fB\-x\fP \fIkeyseq\fP:\fIshell\-command\fP
.TP
\fBbind\fP [\fB\-m\fP \fIkeymap\fP] \fIkeyseq\fP:\fIfunction\-name\fP
.TP
\fBbind\fP \fIreadline\-command\fP
.PD
Display current
.B readline
key and function bindings, bind a key sequence to a
.B readline
function or macro, or set a
.B readline
variable.
Each non-option argument is a command as it would appear in
.IR .inputrc ,
but each binding or command must be passed as a separate argument;
e.g., '"\eC\-x\eC\-r": re\-read\-init\-file'.
Options, if supplied, have the following meanings:
.RS
.PD 0
.TP
.B \-m \fIkeymap\fP
Use
.I keymap
as the keymap to be affected by the subsequent bindings.
Acceptable
.I keymap
names are
\fIemacs, emacs\-standard, emacs\-meta, emacs\-ctlx, vi,
vi\-move, vi\-command\fP, and
.IR vi\-insert .
\fIvi\fP is equivalent to \fIvi\-command\fP; \fIemacs\fP is
equivalent to \fIemacs\-standard\fP.
.TP
.B \-l
List the names of all \fBreadline\fP functions.
.TP
.B \-p
Display \fBreadline\fP function names and bindings in such a way
that they can be re-read.
.TP
.B \-P
List current \fBreadline\fP function names and bindings.
.TP
.B \-v
Display \fBreadline\fP variable names and values in such a way that they
can be re-read.
.TP
.B \-V
List current \fBreadline\fP variable names and values.
```

.TP
.B \-s
Display \fBreadline\fP key sequences bound to macros and the strings
they output in such a way that they can be re-read.
.TP
.B \-S
Display \fBreadline\fP key sequences bound to macros and the strings
they output.
.TP
.B \-f \fIfilename\fP
Read key bindings from \fIfilename\fP.
.TP
.B \-q \fIfunction\fP
Query about which keys invoke the named \fIfunction\fP.
.TP
.B \-u \fIfunction\fP
Unbind all keys bound to the named \fIfunction\fP.
.TP
.B \-r \fIkeyseq\fP
Remove any current binding for \fIkeyseq\fP.
.TP
.B \-x \fIkeyseq\fP:\fIshell\-command\fP
Cause \fIshell\-command\fP to be executed whenever \fIkeyseq\fP is
entered.
.PD
.PP
The return value is 0 unless an unrecognized option is given or an
error occurred.
.RE
.TP
\fBbreak\fP [\fIn\fP]
Exit from within a
.BR for ,
.BR while ,
.BR until ,
or
.B select
loop. If \fIn\fP is specified, break \fIn\fP levels.
.I n
must be \(>= 1. If
.I n
is greater than the number of enclosing loops, all enclosing loops
are exited. The return value is 0 unless the shell is not executing
a loop when
.B break
is executed.
.TP
\fBbuiltin\fP \fIshell\-builtin\fP [\fIarguments\fP]
Execute the specified shell builtin, passing it
.IR arguments ,
and return its exit status.
This is useful when defining a
function whose name is the same as a shell builtin,
retaining the functionality of the builtin within the function.
The \fBcd\fP builtin is commonly redefined this way.
The return status is false if
.I shell\-builtin
is not a shell builtin command.
.TP
\fBcd\fP [\fB\-L|-P\fP] [\fIdir\fP]
Change the current directory to \fIdir\fP. The variable
.SM
.B HOME
is the
default
.IR dir .
The variable
.SM
.B CDPATH
defines the search path for the directory containing
.IR dir .
Alternative directory names in
.SM
.B CDPATH
are separated by a colon (:). A null directory name in
.SM
.B CDPATH
is the same as the current directory, i.e., ``\fB.\fP''. If
.I dir
begins with a slash (/),
then
.SM
.B CDPATH
is not used. The
.B \-P
option says to use the physical directory structure instead of
following symbolic links (see also the
.B \-P
option to the
.B set
builtin command); the
.B \-L
option forces symbolic links to be followed. An argument of
.B \-
is equivalent to
.SM
.BR $OLDPWD .
If a non-empty directory name from \fBCDPATH\fP is used, or if
\fB\-\fP is the first argument, and the directory change is
successful, the absolute pathname of the new working directory is
written to the standard output.
The return value is true if the directory was successfully changed;
false otherwise.
.TP
\fBcaller\fP [\fIexpr\fP]
Returns the context of any active subroutine call (a shell function or
a script executed with the \fB.\fP or \fBsource\fP builtins.
Without \fIexpr\fP, \fBcaller\fP displays the line number and source
filename of the current subroutine call.
If a non-negative integer is supplied as \fIexpr\fP, \fBcaller\fP
displays the line number, subroutine name, and source file corresponding
to that position in the current execution call stack. This extra
information may be used, for example, to print a stack trace. The
current frame is frame 0.
The return value is 0 unless the shell is not executing a subroutine
call or \fIexpr\fP does not correspond to a valid position in the
call stack.
.TP
\fBcommand\fP [\fB\-pVv\fP] \fIcommand\fP [\fIarg\fP ...]
Run
.I command
with

.I args
suppressing the normal shell function lookup. Only builtin
commands or commands found in the
.SM
.B PATH
are executed. If the
.B \-p
option is given, the search for
.I command
is performed using a default value for
.B PATH
that is guaranteed to find all of the standard utilities.
If either the
.B \-V
or
.B \-v
option is supplied, a description of
.I command
is printed. The
.B \-v
option causes a single word indicating the command or file name
used to invoke
.I command
to be displayed; the
.B \-V
option produces a more verbose description.
If the
.B \-V
or
.B \-v
option is supplied, the exit status is 0 if
.I command
was found, and 1 if not. If neither option is supplied and
an error occurred or
.I command
cannot be found, the exit status is 127. Otherwise, the exit status of the
.B command
builtin is the exit status of
.IR command .
.TP
\fBcompgen\fP [\fIoption\fP] [\fIword\fP]
Generate possible completion matches for \fIword\fP according to
the \fIoption\fPs, which may be any option accepted by the
.B complete
builtin with the exception of \fB\-p\fP and \fB\-r\fP, and write
the matches to the standard output.
When using the \fB\-F\fP or \fB\-C\fP options, the various shell variables
set by the programmable completion facilities, while available, will not
have useful values.
.sp 1
The matches will be generated in the same way as if the programmable
completion code had generated them directly from a completion specification
with the same flags.
If \fIword\fP is specified, only those completions matching \fIword\fP
will be displayed.
.sp 1
The return value is true unless an invalid option is supplied, or no
matches were generated.
.TP
\fBcomplete\fP [\fB\-abcdefgjksuv\fP] [\fB\-o\fP \fIcomp-option\fP] [\fB\-A\fP \fIaction\fP] [\fB\-G\fP \fIglobpat\fP] [\fB\-W\fP \fIwordlist\fP] [\fB\-P\fP \
fIprefix\fP] [\fB\-S\fP \fIsuffix\fP]
.br
[\fB\-X\fP \fIfilterpat\fP] [\fB\-F\fP \fIfunction\fP] [\fB\-C\fP \fIcommand\fP] \fIname\fP [\fIname ...\fP]
.PD 0
.TP
\fBcomplete\fP \fB\-pr\fP [\fIname\fP ...]
.PD
Specify how arguments to each \fIname\fP should be completed.
If the \fB\-p\fP option is supplied, or if no options are supplied,
existing completion specifications are printed in a way that allows
them to be reused as input.
The \fB\-r\fP option removes a completion specification for
each \fIname\fP, or, if no \fIname\fPs are supplied, all
completion specifications.
.sp 1
The process of applying these completion specifications when word completion
is attempted is described above under \fBProgrammable Completion\fP.
.sp 1
Other options, if specified, have the following meanings.
The arguments to the \fB\-G\fP, \fB\-W\fP, and \fB\-X\fP options
(and, if necessary, the \fB\-P\fP and \fB\-S\fP options)
should be quoted to protect them from expansion before the
.B complete
builtin is invoked.
.RS
.PD 0
.TP 8
\fB\-o\fP \fIcomp-option\fP
The \fIcomp-option\fP controls several aspects of the compspec's behavior
beyond the simple generation of completions.
\fIcomp-option\fP may be one of:
.RS
.TP 8
.B bashdefault
Perform the rest of the default \fBbash\fP completions if the compspec
generates no matches.
.TP 8
.B default
Use readline's default filename completion if the compspec generates
no matches.
.TP 8
.B dirnames
Perform directory name completion if the compspec generates no matches.
.TP 8
.B filenames
Tell readline that the compspec generates filenames, so it can perform any
filename\-specific processing (like adding a slash to directory names or
suppressing trailing spaces). Intended to be used with shell functions.
.TP 8
.B nospace
Tell readline not to append a space (the default) to words completed at
the end of the line.
.TP 8
.B plusdirs
After any matches defined by the compspec are generated,
directory name completion is attempted and any
matches are added to the results of the other actions.
.RE
.TP 8
\fB\-A\fP \fIaction\fP
The \fIaction\fP may be one of the following to generate a list of possible
completions:

.RS
.TP 8
.B alias
Alias names. May also be specified as \fB\-a\fP.
.TP 8
.B arrayvar
Array variable names.
.TP 8
.B binding
\fBReadline\fP key binding names.
.TP 8
.B builtin
Names of shell builtin commands. May also be specified as \fB\-b\fP.
.TP 8
.B command
Command names. May also be specified as \fB\-c\fP.
.TP 8
.B directory
Directory names. May also be specified as \fB\-d\fP.
.TP 8
.B disabled
Names of disabled shell builtins.
.TP 8
.B enabled
Names of enabled shell builtins.
.TP 8
.B export
Names of exported shell variables. May also be specified as \fB\-e\fP.
.TP 8
.B file
File names. May also be specified as \fB\-f\fP.
.TP 8
.B function
Names of shell functions.
.TP 8
.B group
Group names. May also be specified as \fB\-g\fP.
.TP 8
.B helptopic
Help topics as accepted by the \fBhelp\fP builtin.
.TP 8
.B hostname
Hostnames, as taken from the file specified by the
.SM
.B HOSTFILE
shell variable.
.TP 8
.B job
Job names, if job control is active. May also be specified as \fB\-j\fP.
.TP 8
.B keyword
Shell reserved words. May also be specified as \fB\-k\fP.
.TP 8
.B running
Names of running jobs, if job control is active.
.TP 8
.B service
Service names. May also be specified as \fB\-s\fP.
.TP 8
.B setopt
Valid arguments for the \fB\-o\fP option to the \fBset\fP builtin.
.TP 8
.B shopt
Shell option names as accepted by the \fBshopt\fP builtin.
.TP 8
.B signal
Signal names.
.TP 8
.B stopped
Names of stopped jobs, if job control is active.
.TP 8
.B user
User names. May also be specified as \fB\-u\fP.
.TP 8
.B variable
Names of all shell variables. May also be specified as \fB\-v\fP.
.RE
.TP 8
\fB\-G\fP \fIglobpat\fP
The filename expansion pattern \fIglobpat\fP is expanded to generate
the possible completions.
.TP 8
\fB\-W\fP \fIwordlist\fP
The \fIwordlist\fP is split using the characters in the
.SM
.B IFS
special variable as delimiters, and each resultant word is expanded.
The possible completions are the members of the resultant list which
match the word being completed.
.TP 8
\fB\-C\fP \fIcommand\fP
\fIcommand\fP is executed in a subshell environment, and its output is
used as the possible completions.
.TP 8
\fB\-F\fP \fIfunction\fP
The shell function \fIfunction\fP is executed in the current shell
environment.
When it finishes, the possible completions are retrieved from the value
of the
.SM
.B COMPREPLY
array variable.
.TP 8
\fB\-X\fP \fIfilterpat\fP
\fIfilterpat\fP is a pattern as used for filename expansion.
It is applied to the list of possible completions generated by the
preceding options and arguments, and each completion matching
\fIfilterpat\fP is removed from the list.
A leading \fB!\fP in \fIfilterpat\fP negates the pattern; in this
case, any completion not matching \fIfilterpat\fP is removed.
.TP 8
\fB\-P\fP \fIprefix\fP
\fIprefix\fP is added at the beginning of each possible completion
after all other options have been applied.
.TP 8
\fB\-S\fP \fIsuffix\fP
\fIsuffix\fP is appended to each possible completion
after all other options have been applied.
.PD
.PP
The return value is true unless an invalid option is supplied, an option
other than \fB\-p\fP or \fB\-r\fP is supplied without a \fIname\fP

argument, an attempt is made to remove a completion specification for
a \fIname\fP for which no specification exists, or
an error occurs adding a completion specification.
.RE
.TP
\fBcontinue\fP [\fIn\fP]
Resume the next iteration of the enclosing
.BR for ,
.BR while ,
.BR until ,
or
.B select
loop.
If
.I n
is specified, resume at the \fIn\fPth enclosing loop.
.I n
must be \(>= 1. If
.I n
is greater than the number of enclosing loops, the last enclosing loop
(the ``top-level'' loop) is resumed. The return value is 0 unless the
shell is not executing a loop when
.B continue
is executed.
.TP
\fBdeclare\fP [\fB\-afFirtx\fP] [\fB\-p\fP] [\fIname\fP[=\fIvalue\fP] ...]
.PD 0
.TP
\fBtypeset\fP [\fB\-afFirtx\fP] [\fB\-p\fP] [\fIname\fP[=\fIvalue\fP] ...]
.PD
Declare variables and/or give them attributes.
If no \fIname\fPs are given then display the values of variables.
The
.B \-p
option will display the attributes and values of each
.IR name .
When
.B \-p
is used, additional options are ignored.
The
.B \-F
option inhibits the display of function definitions; only the
function name and attributes are printed.
If the \fBextdebug\fP shell option is enabled using \fBshopt\fP,
the source file name and line number where the function is defined
are displayed as well. The
.B \-F
option implies
.BR \-f .
The following options can
be used to restrict output to variables with the specified attribute or
to give variables attributes:
.RS
.PD 0
.TP
.B \-a
Each \fIname\fP is an array variable (see
.B Arrays
above).
.TP
.B \-f
Use function names only.
.TP
.B \-i
The variable is treated as an integer; arithmetic evaluation (see
.SM
.B "ARITHMETIC EVALUATION" ") "
is performed when the variable is assigned a value.
.TP
.B \-r
Make \fIname\fPs readonly. These names cannot then be assigned values
by subsequent assignment statements or unset.
.TP
.B \-t
Give each \fIname\fP the \fItrace\fP attribute.
Traced functions inherit the \fBDEBUG\fP and \fBRETURN\fP traps from
the calling shell.
The trace attribute has no special meaning for variables.
.TP
.B \-x
Mark \fIname\fPs for export to subsequent commands via the environment.
.PD
.PP
Using `+' instead of `\-'
turns off the attribute instead, with the exception that \fB+a\fP
may not be used to destroy an array variable. When used in a function,
makes each
\fIname\fP local, as with the
.B local
command.
If a variable name is followed by =\fIvalue\fP, the value of
the variable is set to \fIvalue\fP.
The return value is 0 unless an invalid option is encountered,
an attempt is made to define a function using
.if n ``\-f foo=bar'',
.if t \f(CW\-f foo=bar\fP,
an attempt is made to assign a value to a readonly variable,
an attempt is made to assign a value to an array variable without
using the compound assignment syntax (see
.B Arrays
above), one of the \fInames\fP is not a valid shell variable name,
an attempt is made to turn off readonly status for a readonly variable,
an attempt is made to turn off array status for an array variable,
or an attempt is made to display a non-existent function with \fB\-f\fP.
.RE
.TP
.B dirs [\fB\-clpv\fP] [+\fIn\fP] [\-\fIn\fP]
Without options, displays the list of currently remembered directories.
The default display is on a single line with directory names separated
by spaces.
Directories are added to the list with the
.B pushd
command; the
.B popd
command removes entries from the list.
.RS
.PD 0
.TP
\fB+\fP\fIn\fP
Displays the \fIn\fPth entry counting from the left of the list
shown by
.B dirs

when invoked without options, starting with zero.
.TP
\fB\-\fP\fIn\fP
Displays the \fIn\fPth entry counting from the right of the list
shown by
.B dirs
when invoked without options, starting with zero.
.TP
.B \-c
Clears the directory stack by deleting all of the entries.
.TP
.B \-l
Produces a longer listing; the default listing format uses a
tilde to denote the home directory.
.TP
.B \-p
Print the directory stack with one entry per line.
.TP
.B \-v
Print the directory stack with one entry per line,
prefixing each entry with its index in the stack.
.PD
.PP
The return value is 0 unless an
invalid option is supplied or \fIn\fP indexes beyond the end
of the directory stack.
.RE
.TP
\fBdisown\fP [\fB\-ar\fP] [\fB\-h\fP] [\fIjobspec\fP ...]
Without options, each
.I jobspec
is removed from the table of active jobs.
If the \fB\-h\fP option is given, each
.I jobspec
is not removed from the table, but is marked so that
.SM
.B SIGHUP
is not sent to the job if the shell receives a
.SM
.BR SIGHUP .
If no
.I jobspec
is present, and neither the
.B \-a
nor the
.B \-r
option is supplied, the \fIcurrent job\fP is used.
If no
.I jobspec
is supplied, the
.B \-a
option means to remove or mark all jobs; the
.B \-r
option without a
.I jobspec
argument restricts operation to running jobs.
The return value is 0 unless a
.I jobspec
does not specify a valid job.
.TP
\fBecho\fP [\fB\-neE\fP] [\fIarg\fP ...]
Output the \fIarg\fPs, separated by spaces, followed by a newline.
The return status is always 0.
If \fB\-n\fP is specified, the trailing newline is
suppressed. If the \fB\-e\fP option is given, interpretation of
the following backslash-escaped characters is enabled. The
.B \-E
option disables the interpretation of these escape characters,
even on systems where they are interpreted by default.
The \fBxpg_echo\fP shell option may be used to
dynamically determine whether or not \fBecho\fP expands these
escape characters by default.
.B echo
does not interpret \fB\-\-\fP to mean the end of options.
.B echo
interprets the following escape sequences:
.RS
.PD 0
.TP
.B \ea
alert (bell)
.TP
.B \eb
backspace
.TP
.B \ec
suppress trailing newline
.TP
.B \ee
an escape character
.TP
.B \ef
form feed
.TP
.B \en
new line
.TP
.B \er
carriage return
.TP
.B \et
horizontal tab
.TP
.B \ev
vertical tab
.TP
.B \e\e
backslash
.TP
.B \e0\fInnn\fP
the eight-bit character whose value is the octal value \fInnn\fP
(zero to three octal digits)
.TP
.B \ex\fIHH\fP
the eight-bit character whose value is the hexadecimal value \fIHH\fP
(one or two hex digits)
.PD
.RE
.TP
\fBenable\fP [\fB\-adnps\fP] [\fB\-f\fP \fIfilename\fP] [\fIname\fP ...]
Enable and disable builtin shell commands.
Disabling a builtin allows a disk command which has the same name

as a shell builtin to be executed without specifying a full pathname,
even though the shell normally searches for builtins before disk commands.
If \fB\-n\fP is used, each \fIname\fP
is disabled; otherwise,
\fInames\fP are enabled. For example, to use the
.B test
binary found via the
.SM
.B PATH
instead of the shell builtin version, run
.if t \f(CWenable -n test\fP.
.if n ``enable -n test''.
The
.B \-f
option means to load the new builtin command
.I name
from shared object
.IR filename ,
on systems that support dynamic loading. The
.B \-d
option will delete a builtin previously loaded with
.BR \-f .
If no \fIname\fP arguments are given, or if the
.B \-p
option is supplied, a list of shell builtins is printed.
With no other option arguments, the list consists of all enabled
shell builtins.
If \fB\-n\fP is supplied, only disabled builtins are printed.
If \fB\-a\fP is supplied, the list printed includes all builtins, with an
indication of whether or not each is enabled.
If \fB\-s\fP is supplied, the output is restricted to the POSIX
\fIspecial\fP builtins.
The return value is 0 unless a
.I name
is not a shell builtin or there is an error loading a new builtin
from a shared object.
.TP
\fBeval\fP [\fIarg\fP ...]
The \fIarg\fPs are read and concatenated together into a single
command. This command is then read and executed by the shell, and
its exit status is returned as the value of
.BR eval .
If there are no
.IR args ,
or only null arguments,
.B eval
returns 0.
.TP
\fBexec\fP [\fB\-cl\fP] [\fB\-a\fP \fIname\fP] [\fIcommand\fP [\fIarguments\fP]]
If
.I command
is specified, it replaces the shell.
No new process is created. The
.I arguments
become the arguments to \fIcommand\fP.
If the
.B \-l
option is supplied,
the shell places a dash at the beginning of the zeroth arg passed to
.IR command .
This is what
.IR login (1)
does. The
.B \-c
option causes
.I command
to be executed with an empty environment. If
.B \-a
is supplied, the shell passes
.I name
as the zeroth argument to the executed command. If
.I command
cannot be executed for some reason, a non-interactive shell exits,
unless the shell option
.B execfail
is enabled, in which case it returns failure.
An interactive shell returns failure if the file cannot be executed.
If
.I command
is not specified, any redirections take effect in the current shell,
and the return status is 0. If there is a redirection error, the
return status is 1.
.TP
\fBexit\fP [\fIn\fP]
Cause the shell to exit
with a status of \fIn\fP. If
.I n
is omitted, the exit status
is that of the last command executed.
A trap on
.SM
.B EXIT
is executed before the shell terminates.
.TP
\fBexport\fP [\fB\-fn\fP\^] [\fIname\fP[=\fIword\fP]] ...
.PD 0
.TP
.B export \-p
.PD
The supplied
.I names
are marked for automatic export to the environment of
subsequently executed commands. If the
.B \-f
option is given,
the
.I names
refer to functions.
If no
.I names
are given, or if the
.B \-p
option is supplied, a list
of all names that are exported in this shell is printed.
The
.B \-n
option causes the export property to be removed from each
\fIname\fP.
If a variable name is followed by =\fIword\fP, the value of
the variable is set to \fIword\fP.
.B export
returns an exit status of 0 unless an invalid option is

encountered,
one of the \fInames\fP is not a valid shell variable name, or
.B \-f
is supplied with a
.I name
that is not a function.
.TP
\fBfc\fP [\fB\-e\fP \fIename\fP] [\fB\-nlr\fP] [\fIfirst\fP] [\fIlast\fP]
.PD 0
.TP
\fBfc\fP \fB\-s\fP [\fIpat\fP=\fIrep\fP] [\fIcmd\fP]
.PD
Fix Command. In the first form, a range of commands from
.I first
to
.I last
is selected from the history list.
.I First
and
.I last
may be specified as a string (to locate the last command beginning
with that string) or as a number (an index into the history list,
where a negative number is used as an offset from the current
command number). If
.I last
is not specified it is set to
the current command for listing (so that
.if n ``fc \-l \-10''
.if t \f(CWfc \-l \-10\fP
prints the last 10 commands) and to
.I first
otherwise.
If
.I first
is not specified it is set to the previous
command for editing and \-16 for listing.
.sp 1
The
.B \-n
option suppresses
the command numbers when listing. The
.B \-r
option reverses the order of
the commands. If the
.B \-l
option is given,
the commands are listed on
standard output. Otherwise, the editor given by
.I ename
is invoked
on a file containing those commands. If
.I ename
is not given, the
value of the
.SM
.B FCEDIT
variable is used, and
the value of
.SM
.B EDITOR
if
.SM
.B FCEDIT
is not set. If neither variable is set,
.FN vi
is used. When editing is complete, the edited commands are
echoed and executed.
.sp 1
In the second form, \fIcommand\fP is re-executed after each instance
of \fIpat\fP is replaced by \fIrep\fP.
A useful alias to use with this is
.if n ``r="fc -s"'',
.if t \f(CWr='fc \-s'\fP,
so that typing
.if n ``r cc''
.if t \f(CWr cc\fP
runs the last command beginning with
.if n ``cc''
.if t \f(CWcc\fP
and typing
.if n ``r''
.if t \f(CWr\fP
re-executes the last command.
.sp 1
If the first form is used, the return value is 0 unless an invalid
option is encountered or
.I first
or
.I last
specify history lines out of range.
If the
.B \-e
option is supplied, the return value is the value of the last
command executed or failure if an error occurs with the temporary
file of commands. If the second form is used, the return status
is that of the command re-executed, unless
.I cmd
does not specify a valid history line, in which case
.B fc
returns failure.
.TP
\fBfg\fP [\fIjobspec\fP]
Resume
.I jobspec
in the foreground, and make it the current job.
If
.I jobspec
is not present, the shell's notion of the \fIcurrent job\fP is used.
The return value is that of the command placed into the foreground,
or failure if run when job control is disabled or, when run with
job control enabled, if
.I jobspec
does not specify a valid job or
.I jobspec
specifies a job that was started without job control.
.TP
\fBgetopts\fP \fIoptstring\fP \fIname\fP [\fIargs\fP]
.B getopts
is used by shell procedures to parse positional parameters.
.I optstring
contains the option characters to be recognized; if a character
is followed by a colon, the option is expected to have an

argument, which should be separated from it by white space.
The colon and question mark characters may not be used as
option characters.
Each time it is invoked,
.B getopts
places the next option in the shell variable
.IR name ,
initializing
.I name
if it does not exist,
and the index of the next argument to be processed into the
variable
.SM
.BR OPTIND .
.SM
.B OPTIND
is initialized to 1 each time the shell or a shell script
is invoked. When an option requires an argument,
.B getopts
places that argument into the variable
.SM
.BR OPTARG .
The shell does not reset
.SM
.B OPTIND
automatically; it must be manually reset between multiple
calls to
.B getopts
within the same shell invocation if a new set of parameters
is to be used.
.sp 1
When the end of options is encountered, \fBgetopts\fP exits with a
return value greater than zero.
\fBOPTIND\fP is set to the index of the first non-option argument,
and \fBname\fP is set to ?.
.sp 1
.B getopts
normally parses the positional parameters, but if more arguments are
given in
.IR args ,
.B getopts
parses those instead.
.sp 1
.B getopts
can report errors in two ways. If the first character of
.I optstring
is a colon,
.I silent
error reporting is used. In normal operation diagnostic messages
are printed when invalid options or missing option arguments are
encountered.
If the variable
.SM
.B OPTERR
is set to 0, no error messages will be displayed, even if the first
character of
.I optstring
is not a colon.
.sp 1
If an invalid option is seen,
.B getopts
places ? into
.I name
and, if not silent,
prints an error message and unsets
.SM
.BR OPTARG .
If
.B getopts
is silent,
the option character found is placed in
.SM
.B OPTARG
and no diagnostic message is printed.
.sp 1
If a required argument is not found, and
.B getopts
is not silent,
a question mark (\^\fB?\fP\^) is placed in
.IR name ,
.SM
.B OPTARG
is unset, and a diagnostic message is printed.
If
.B getopts
is silent, then a colon (\^\fB:\fP\^) is placed in
.I name
and
.SM
.B OPTARG
is set to the option character found.
.sp 1
.B getopts
returns true if an option, specified or unspecified, is found.
It returns false if the end of options is encountered or an
error occurs.
.TP
\fBhash\fP [\fB\-lr\fP] [\fB\-p\fP \fIfilename\fP] [\fB\-dt\fP] [\fIname\fP]
For each
.IR name ,
the full file name of the command is determined by searching
the directories in
.B $PATH
and remembered.
If the
.B \-p
option is supplied, no path search is performed, and
.I filename
is used as the full file name of the command.
The
.B \-r
option causes the shell to forget all
remembered locations.
The
.B \-d
option causes the shell to forget the remembered location of each \fIname\fP.
If the
.B \-t
option is supplied, the full pathname to which each \fIname\fP corresponds
is printed. If multiple \fIname\fP arguments are supplied with \fB\-t\fP,
the \fIname\fP is printed before the hashed full pathname.
The

```
.B \-l
option causes output to be displayed in a format that may be reused as input.
If no arguments are given, or if only \fB\-l\fP is supplied,
information about remembered commands is printed.
The return status is true unless a
.I name
is not found or an invalid option is supplied.
.TP
\fBhelp\fP [\fB\-s\fP] [\fIpattern\fP]
Display helpful information about builtin commands.  If
.I pattern
is specified,
.B help
gives detailed help on all commands matching
.IR pattern ;
otherwise help for all the builtins and shell control structures
is printed.
The \fB\-s\fP option restricts the information displayed to a short
usage synopsis.
The return status is 0 unless no command matches
.IR pattern .
.TP
\fBhistory [\fIn\fP]
.PD 0
.TP
\fBhistory\fP \fB\-c\fP
.TP
\fBhistory \-d\fP \fIoffset\fP
.TP
\fBhistory\fP \fB\-anrw\fP [\fIfilename\fP]
.TP
\fBhistory\fP \fB\-p\fP \fIarg\fP [\fIarg ...\fP]
.TP
\fBhistory\fP \fB\-s\fP \fIarg\fP [\fIarg ...\fP]
.PD
With no options, display the command
history list with line numbers.  Lines listed
with a
.B *
have been modified.  An argument of
.I n
lists only the last
.I n
lines.
If the shell variable \fBHISTTIMEFORMAT\fP is set and not null,
it is used as a format string for \fIstrftime\fP(3) to display
the time stamp associated with each displayed history entry.
No intervening blank is printed between the formatted time stamp
and the history line.
If \fIfilename\fP is supplied, it is used as the
name of the history file; if not, the value of
.SM
.B HISTFILE
is used.  Options, if supplied, have the following meanings:
.RS
.PD 0
.TP
.B \-c
Clear the history list by deleting all the entries.
.TP
\fB\-d\fP \fIoffset\fP
Delete the history entry at position \fIoffset\fP.
.TP
.B \-a
Append the ``new'' history lines (history lines entered since the
beginning of the current \fBbash\fP session) to the history file.
.TP
.B \-n
Read the history lines not already read from the history
file into the current history list.  These are lines
appended to the history file since the beginning of the
current \fBbash\fP session.
.TP
.B \-r
Read the contents of the history file
and use them as the current history.
.TP
.B \-w
Write the current history to the history file, overwriting the
history file's contents.
.TP
.B \-p
Perform history substitution on the following \fIargs\fP and display
the result on the standard output.
Does not store the results in the history list.
Each \fIarg\fP must be quoted to disable normal history expansion.
.TP
.B \-s
Store the
.I args
in the history list as a single entry.  The last command in the
history list is removed before the
.I args
are added.
.PD
.PP
If the \fBHISTTIMEFORMAT\fP is set, the time stamp information
associated with each history entry is written to the history file.
The return value is 0 unless an invalid option is encountered, an
error occurs while reading or writing the history file, an invalid
\fIoffset\fP is supplied as an argument to \fB\-d\fP, or the
history expansion supplied as an argument to \fB\-p\fP fails.
.RE
.TP
\fBjobs\fP [\fB\-lnprs\fP] [ \fIjobspec\fP ... ]
.PD 0
.TP
\fBjobs\fP \fB\-x\fP \fIcommand\fP [ \fIargs\fP ... ]
.PD
The first form lists the active jobs.  The options have the following
meanings:
.RS
.PD 0
.TP
.B \-l
List process IDs
in addition to the normal information.
.TP
.B \-p
List only the process ID of the job's process group
leader.
.TP
```

.B \-n
Display information only about jobs that have changed status since
the user was last notified of their status.
.TP
.B \-r
Restrict output to running jobs.
.TP
.B \-s
Restrict output to stopped jobs.
.PD
.PP
If
.I jobspec
is given, output is restricted to information about that job.
The return status is 0 unless an invalid option is encountered
or an invalid
.I jobspec
is supplied.
.PP
If the
.B \-x
option is supplied,
.B jobs
replaces any
.I jobspec
found in
.I command
or
.I args
with the corresponding process group ID, and executes
.I command
passing it
.IR args ,
returning its exit status.
.RE
.TP
\fBkill\fP [\fB\-s\fP \fIsigspec\fP | \fB\-n\fP \fIsignum\fP | \fB\-\fP\fIsigspec\fP] [\fIpid\fP | \fIjobspec\fP] ...
.PD 0
.TP
\fBkill\fP \fB\-l\fP [\fIsigspec\fP | \fIexit_status\fP]
.PD
Send the signal named by
.I sigspec
or
.I signum
to the processes named by
.I pid
or
.IR jobspec .
.I sigspec
is either a case-insensitive signal name such as
.SM
.B SIGKILL
(with or without the
.SM
.B SIG
prefix) or a signal number;
.I signum
is a signal number.
If
.I sigspec
is not present, then
.SM
.B SIGTERM
is assumed.
An argument of
.B \-l
lists the signal names.
If any arguments are supplied when
.B \-l
is given, the names of the signals corresponding to the arguments are
listed, and the return status is 0.
The \fIexit_status\fP argument to
.B \-l
is a number specifying either a signal number or the exit status of
a process terminated by a signal.
.B kill
returns true if at least one signal was successfully sent, or false
if an error occurs or an invalid option is encountered.
.TP
\fBlet\fP \fIarg\fP [\fIarg\fP ...]
Each
.I arg
is an arithmetic expression to be evaluated (see
.SM
.BR "ARITHMETIC EVALUATION").
If the last
.I arg
evaluates to 0,
.B let
returns 1; 0 is returned otherwise.
.TP
\fBlocal\fP [\fIoption\fP] [\fIname\fP[=\fIvalue\fP] ...]
For each argument, a local variable named
.I name
is created, and assigned
.IR value .
The \fIoption\fP can be any of the options accepted by \fBdeclare\fP.
When
.B local
is used within a function, it causes the variable
.I name
to have a visible scope restricted to that function and its children.
With no operands,
.B local
writes a list of local variables to the standard output. It is
an error to use
.B local
when not within a function. The return status is 0 unless
.B local
is used outside a function, an invalid
.I name
is supplied, or
\fIname\fP is a readonly variable.
.TP
.B logout
Exit a login shell.
.TP
\fBpopd\fP [\-\fBn\fP] [+\fIn\fP] [\-\fIn\fP]
Removes entries from the directory stack. With no arguments,
removes the top directory from the stack, and performs a
.B cd

```
to the new top directory.
Arguments, if supplied, have the following meanings:
.RS
.PD 0
.TP
\fB+\fP\fIn\fP
Removes the \fIn\fPth entry counting from the left of the list
shown by
.BR dirs ,
starting with zero.  For example:
.if n ``popd +0''
.if t \f(CWpopd +0\fP
removes the first directory,
.if n ``popd +1''
.if t \f(CWpopd +1\fP
the second.
.TP
\fB\-\fP\fIn\fP
Removes the \fIn\fPth entry counting from the right of the list
shown by
.BR dirs ,
starting with zero.  For example:
.if n ``popd -0''
.if t \f(CWpopd -0\fP
removes the last directory,
.if n ``popd -1''
.if t \f(CWpopd -1\fP
the next to last.
.TP
.B \-n
Suppresses the normal change of directory when removing directories
from the stack, so that only the stack is manipulated.
.PD
.PP
If the
.B popd
command is successful, a
.B dirs
is performed as well, and the return status is 0.
.B popd
returns false if an invalid option is encountered, the directory stack
is empty, a non-existent directory stack entry is specified, or the
directory change fails.
.RE
.TP
\fBprintf\fP [\fB\-v\fP \fIvar\fP] \fIformat\fP [\fIarguments\fP]
Write the formatted \fIarguments\fP to the standard output under the
control of the \fIformat\fP.
The \fIformat\fP is a character string which contains three types of objects:
plain characters, which are simply copied to standard output, character
escape sequences, which are converted and copied to the standard output, and
format specifications, each of which causes printing of the next successive
\fIargument\fP.
In addition to the standard \fIprintf\fP(1) formats, \fB%b\fP causes
\fBprintf\fP to expand backslash escape sequences in the corresponding
\fIargument\fP (except that \fB\ec\fP terminates output, backslashes in
\fB\e\(aq\fP, \fB\e"\fP, and \fB\e?\fP are not removed, and octal escapes
beginning with \fB\e0\fP may contain up to four digits),
and \fB%q\fP causes \fBprintf\fP to output the corresponding
\fIargument\fP in a format that can be reused as shell input.
.sp 1
The \fB\-v\fP option causes the output to be assigned to the variable
\fIvar\fP rather than being printed to the standard output.
.sp 1
The \fIformat\fP is reused as necessary to consume all of the \fIarguments\fP.
If the \fIformat\fP requires more \fIarguments\fP than are supplied, the
extra format specifications behave as if a zero value or null string, as
appropriate, had been supplied.  The return value is zero on success,
non-zero on failure.
.TP
\fBpushd\fP [\fB\-n\fP] [\fIdir\fP]
.PD 0
.TP
\fBpushd\fP [\fB\-n\fP] [+\fIn\fP] [\-\fIn\fP]
.PD
Adds a directory to the top of the directory stack, or rotates
the stack, making the new top of the stack the current working
directory.  With no arguments, exchanges the top two directories
and returns 0, unless the directory stack is empty.
Arguments, if supplied, have the following meanings:
.RS
.PD 0
.TP
\fB+\fP\fIn\fP
Rotates the stack so that the \fIn\fPth directory
(counting from the left of the list shown by
.BR dirs ,
starting with zero)
is at the top.
.TP
\fB\-\fP\fIn\fP
Rotates the stack so that the \fIn\fPth directory
(counting from the right of the list shown by
.BR dirs ,
starting with zero) is at the top.
.TP
.B \-n
Suppresses the normal change of directory when adding directories
to the stack, so that only the stack is manipulated.
.TP
.I dir
Adds
.I dir
to the directory stack at the top, making it the
new current working directory.
.PD
.PP
If the
.B pushd
command is successful, a
.B dirs
is performed as well.
If the first form is used,
.B pushd
returns 0 unless the cd to
.I dir
fails.  With the second form,
.B pushd
returns 0 unless the directory stack is empty,
a non-existent directory stack element is specified,
or the directory change to the specified new current directory
fails.
```

.RE
.TP
\fBpwd\fP [\fB\-LP\fP]
Print the absolute pathname of the current working directory.
The pathname printed contains no symbolic links if the
.B \-P
option is supplied or the
.B \-o physical
option to the
.B set
builtin command is enabled.
If the
.B \-L
option is used, the pathname printed may contain symbolic links.
The return status is 0 unless an error occurs while
reading the name of the current directory or an
invalid option is supplied.
.TP
\fBread\fP [\fB\-ers\fP] [\fB\-u\fP \fIfd\fP] [\fB\-t\fP \fItimeout\fP] [\fB\-a\fP \fIaname\fP] [\fB\-p\fP \fIprompt\fP] [\fB\-n\fP \fInchars\fP] [\fB\-d\fP \
fIdelim\fP] [\fIname\fP ...]
One line is read from the standard input, or from the file descriptor
\fIfd\fP supplied as an argument to the \fB\-u\fP option, and the first word
is assigned to the first
.IR name ,
the second word to the second
.IR name ,
and so on, with leftover words and their intervening separators assigned
to the last
.IR name .
If there are fewer words read from the input stream than names,
the remaining names are assigned empty values.
The characters in
.SM
.B IFS
are used to split the line into words.
The backslash character (\fB\e\fP) may be used to remove any special
meaning for the next character read and for line continuation.
Options, if supplied, have the following meanings:
.RS
.PD 0
.TP
.B \-a \fIaname\fP
The words are assigned to sequential indices
of the array variable
.IR aname ,
starting at 0.
.I aname
is unset before any new values are assigned.
Other \fIname\fP arguments are ignored.
.TP
.B \-d \fIdelim\fP
The first character of \fIdelim\fP is used to terminate the input line,
rather than newline.
.TP
.B \-e
If the standard input
is coming from a terminal,
.B readline
(see
.SM
.B READLINE
above) is used to obtain the line.
.TP
.B \-n \fInchars\fP
\fBread\fP returns after reading \fInchars\fP characters rather than
waiting for a complete line of input.
.TP
.B \-p \fIprompt\fP
Display \fIprompt\fP on standard error, without a
trailing newline, before attempting to read any input. The prompt
is displayed only if input is coming from a terminal.
.TP
.B \-r
Backslash does not act as an escape character.
The backslash is considered to be part of the line.
In particular, a backslash-newline pair may not be used as a line
continuation.
.TP
.B \-s
Silent mode. If input is coming from a terminal, characters are
not echoed.
.TP
.B \-t \fItimeout\fP
Cause \fBread\fP to time out and return failure if a complete line of
input is not read within \fItimeout\fP seconds.
This option has no effect if \fBread\fP is not reading input from the
terminal or a pipe.
.TP
.B \-u \fIfd\fP
Read input from file descriptor \fIfd\fP.
.PD
.PP
If no
.I names
are supplied, the line read is assigned to the variable
.SM
.BR REPLY .
The return code is zero, unless end-of-file is encountered, \fBread\fP
times out, or an invalid file descriptor is supplied as the argument to
\fB\-u\fP.
.RE
.TP
\fBreadonly\fP [\fB\-apf\fP] [\fIname\fP[=\fIword\fP] ...]
.PD
The given
\fInames\fP are marked readonly; the values of these
.I names
may not be changed by subsequent assignment.
If the
.B \-f
option is supplied, the functions corresponding to the
\fInames\fP are so
marked.
The
.B \-a
option restricts the variables to arrays.
If no
.I name
arguments are given, or if the
.B \-p
option is supplied, a list of all readonly names is printed.
The

.B \-p
option causes output to be displayed in a format that
may be reused as input.
If a variable name is followed by =\fIword\fP, the value of
the variable is set to \fIword\fP.
The return status is 0 unless an invalid option is encountered,
one of the
.I names
is not a valid shell variable name, or
.B \-f
is supplied with a
.I name
that is not a function.
.TP
\fBreturn\fP [\fIn\fP]
Causes a function to exit with the return value specified by
.IR n .
If
.I n
is omitted, the return status is that of the last command
executed in the function body. If used outside a function,
but during execution of a script by the
.B .
(\fBsource\fP) command, it causes the shell to stop executing
that script and return either
.I n
or the exit status of the last command executed within the
script as the exit status of the script. If used outside a
function and not during execution of a script by \fB.\fP\^,
the return status is false.
Any command associated with the \fBRETURN\fP trap is executed
before execution resumes after the function or script.
.TP
\fBset\fP [\fB\-\-abefhkmnptuvxBCHP\fP] [\fB\-o\fP \fIoption\fP] [\fIarg\fP ...]
Without options, the name and value of each shell variable are displayed
in a format that can be reused as input
for setting or resetting the currently-set variables.
Read-only variables cannot be reset.
In \fIposix mode\fP, only shell variables are listed.
The output is sorted according to the current locale.
When options are specified, they set or unset shell attributes.
Any arguments remaining after the options are processed are treated
as values for the positional parameters and are assigned, in order, to
.BR $1 ,
.BR $2 ,
.B ...
.BR $\fIn\fP .
Options, if specified, have the following meanings:
.RS
.PD 0
.TP 8
.B \-a
Automatically mark variables and functions which are modified or
created for export to the environment of subsequent commands.
.TP 8
.B \-b
Report the status of terminated background jobs
immediately, rather than before the next primary prompt. This is
effective only when job control is enabled.
.TP 8
.B \-e
Exit immediately if a \fIsimple command\fP (see
.SM
.B SHELL GRAMMAR
above) exits with a non-zero status.
The shell does not exit if the
command that fails is part of the command list immediately following a
.B while
or
.B until
keyword,
part of the test in an
.I if
statement, part of a
.B &&
or
.B \(bv\(bv
list, or if the command's return value is
being inverted via
.BR ! .
A trap on \fBERR\fP, if set, is executed before the shell exits.
.TP 8
.B \-f
Disable pathname expansion.
.TP 8
.B \-h
Remember the location of commands as they are looked up for execution.
This is enabled by default.
.TP 8
.B \-k
All arguments in the form of assignment statements
are placed in the environment for a command, not just
those that precede the command name.
.TP 8
.B \-m
Monitor mode. Job control is enabled. This option is on
by default for interactive shells on systems that support
it (see
.SM
.B JOB CONTROL
above). Background processes run in a separate process
group and a line containing their exit status is printed
upon their completion.
.TP 8
.B \-n
Read commands but do not execute them. This may be used to
check a shell script for syntax errors. This is ignored by
interactive shells.
.TP 8
.B \-o \fIoption\-name\fP
The \fIoption\-name\fP can be one of the following:
.RS
.TP 8
.B allexport
Same as
.BR \-a .
.TP 8
.B braceexpand
Same as
.BR \-B .
.TP 8
.B emacs

Use an emacs-style command line editing interface. This is enabled
by default when the shell is interactive, unless the shell is started
with the
.B \-\-noediting
option.
.TP 8
.B errtrace
Same as
.BR \-E .
.TP 8
.B functrace
Same as
.BR \-T .
.TP 8
.B errexit
Same as
.BR \-e .
.TP 8
.B hashall
Same as
.BR \-h .
.TP 8
.B histexpand
Same as
.BR \-H .
.TP 8
.B history
Enable command history, as described above under
.SM
.BR HISTORY .
This option is on by default in interactive shells.
.TP 8
.B ignoreeof
The effect is as if the shell command
.if t \f(CWIGNOREEOF=10\fP
.if n ``IGNOREEOF=10''
had been executed
(see
.B Shell Variables
above).
.TP 8
.B keyword
Same as
.BR \-k .
.TP 8
.B monitor
Same as
.BR \-m .
.TP 8
.B noclobber
Same as
.BR \-C .
.TP 8
.B noexec
Same as
.BR \-n .
.TP 8
.B noglob
Same as
.BR \-f .
.B nolog
Currently ignored.
.TP 8
.B notify
Same as
.BR \-b .
.TP 8
.B nounset
Same as
.BR \-u .
.TP 8
.B onecmd
Same as
.BR \-t .
.TP 8
.B physical
Same as
.BR \-P .
.TP 8
.B pipefail
If set, the return value of a pipeline is the value of the last
(rightmost) command to exit with a non-zero status, or zero if all
commands in the pipeline exit successfully.
This option is disabled by default.
.TP 8
.B posix
Change the behavior of
.B bash
where the default operation differs
from the POSIX standard to match the standard (\fIposix mode\fP).
.TP 8
.B privileged
Same as
.BR \-p .
.TP 8
.B verbose
Same as
.BR \-v .
.TP 8
.B vi
Use a vi-style command line editing interface.
.TP 8
.B xtrace
Same as
.BR \-x .
.sp .5
.PP
If
.B \-o
is supplied with no \fIoption\-name\fP, the values of the current options are
printed.
If
.B +o
is supplied with no \fIoption\-name\fP, a series of
.B set
commands to recreate the current option settings is displayed on
the standard output.
.RE
.TP 8
.B \-p
Turn on
.I privileged

mode. In this mode, the
.SM
.B $ENV
and
.SM
.B $BASH_ENV
files are not processed, shell functions are not inherited from the
environment, and the
.SM
.B SHELLOPTS
variable, if it appears in the environment, is ignored.
If the shell is started with the effective user (group) id not equal to the
real user (group) id, and the \fB\-p\fP option is not supplied, these actions
are taken and the effective user id is set to the real user id.
If the \fB\-p\fP option is supplied at startup, the effective user id is
not reset.
Turning this option off causes the effective user
and group ids to be set to the real user and group ids.
.TP 8
.B \-t
Exit after reading and executing one command.
.TP 8
.B \-u
Treat unset variables as an error when performing
parameter expansion. If expansion is attempted on an
unset variable, the shell prints an error message, and,
if not interactive, exits with a non-zero status.
.TP 8
.B \-v
Print shell input lines as they are read.
.TP 8
.B \-x
After expanding each \fIsimple command\fP,
\fBfor\fP command, \fBcase\fP command, \fBselect\fP command, or
arithmetic \fBfor\fP command, display the expanded value of
.SM
.BR PS4 ,
followed by the command and its expanded arguments
or associated word list.
.TP 8
.B \-B
The shell performs brace expansion (see
.B Brace Expansion
above). This is on by default.
.TP 8
.B \-C
If set,
.B bash
does not overwrite an existing file with the
.BR > ,
.BR >& ,
and
.B <>
redirection operators. This may be overridden when
creating output files by using the redirection operator
.B >|
instead of
.BR > .
.TP 8
.B \-E
If set, any trap on \fBERR\fP is inherited by shell functions, command
substitutions, and commands executed in a subshell environment.
The \fBERR\fP trap is normally not inherited in such cases.
.TP 8
.B \-H
Enable
.B !
style history substitution. This option is on by
default when the shell is interactive.
.TP 8
.B \-P
If set, the shell does not follow symbolic links when executing
commands such as
.B cd
that change the current working directory. It uses the
physical directory structure instead. By default,
.B bash
follows the logical chain of directories when performing commands
which change the current directory.
.TP 8
.B \-T
If set, any traps on \fBDEBUG\fP and \fBRETURN\fP are inherited by shell
functions, command substitutions, and commands executed in a
subshell environment.
The \fBDEBUG\fP and \fBRETURN\fP traps are normally not inherited
in such cases.
.TP 8
.B \-\-
If no arguments follow this option, then the positional parameters are
unset. Otherwise, the positional parameters are set to the
\fIarg\fPs, even if some of them begin with a
.BR \- .
.TP 8
.B \-
Signal the end of options, cause all remaining \fIarg\fPs to be
assigned to the positional parameters. The
.B \-x
and
.B \-v
options are turned off.
If there are no \fIarg\fPs,
the positional parameters remain unchanged.
.PD
.PP
The options are off by default unless otherwise noted.
Using + rather than \- causes these options to be turned off.
The options can also be specified as arguments to an invocation of
the shell.
The current set of options may be found in
.BR $\- .
The return status is always true unless an invalid option is encountered.
.RE
.TP
\fBshift\fP [\fIn\fP]
The positional parameters from \fIn\fP+1 ... are renamed to
.B $1
.B ....
Parameters represented by the numbers \fB$#\fP
down to \fB$#\fP\-\fIn\fP+1 are unset.
.I n
must be a non-negative number less than or equal to \fB$#\fP.
If

.I n
is 0, no parameters are changed.
If
.I n
is not given, it is assumed to be 1.
If
.I n
is greater than \fB$#\fP, the positional parameters are not changed.
The return status is greater than zero if
.I n
is greater than
.B $#
or less than zero; otherwise 0.
.TP
\fBshopt\fP [\fB\-pqsu\fP] [\fB\-o\fP] [\fIoptname\fP ...]
Toggle the values of variables controlling optional shell behavior.
With no options, or with the
.B \-p
option, a list of all settable options is displayed, with
an indication of whether or not each is set.
The \fB\-p\fP option causes output to be displayed in a form that
may be reused as input.
Other options have the following meanings:
.RS
.PD 0
.TP
.B \-s
Enable (set) each \fIoptname\fP.
.TP
.B \-u
Disable (unset) each \fIoptname\fP.
.TP
.B \-q
Suppresses normal output (quiet mode); the return status indicates
whether the \fIoptname\fP is set or unset.
If multiple \fIoptname\fP arguments are given with
.BR \-q ,
the return status is zero if all \fIoptnames\fP are enabled; non-zero
otherwise.
.TP
.B \-o
Restricts the values of \fIoptname\fP to be those defined for the
.B \-o
option to the
.B set
builtin.
.PD
.PP
If either
.B \-s
or
.B \-u
is used with no \fIoptname\fP arguments, the display is limited to
those options which are set or unset, respectively.
Unless otherwise noted, the \fBshopt\fP options are disabled (unset)
by default.
.PP
The return status when listing options is zero if all \fIoptnames\fP
are enabled, non-zero otherwise. When setting or unsetting options,
the return status is zero unless an \fIoptname\fP is not a valid shell
option.
.PP
The list of \fBshopt\fP options is:
.if t .sp .5v
.if n .sp 1v
.PD 0
.TP 8
.B cdable_vars
If set, an argument to the
.B cd
builtin command that
is not a directory is assumed to be the name of a variable whose
value is the directory to change to.
.TP 8
.B cdspell
If set, minor errors in the spelling of a directory component in a
.B cd
command will be corrected.
The errors checked for are transposed characters,
a missing character, and one character too many.
If a correction is found, the corrected file name is printed,
and the command proceeds.
This option is only used by interactive shells.
.TP 8
.B checkhash
If set, \fBbash\fP checks that a command found in the hash
table exists before trying to execute it. If a hashed command no
longer exists, a normal path search is performed.
.TP 8
.B checkwinsize
If set, \fBbash\fP checks the window size after each command
and, if necessary, updates the values of
.SM
.B LINES
and
.SM
.BR COLUMNS .
.TP 8
.B cmdhist
If set,
.B bash
attempts to save all lines of a multiple-line
command in the same history entry. This allows
easy re-editing of multi-line commands.
.TP 8
.B compat31
If set,
.B bash
changes its behavior to that of version 3.1 with respect to quoted
arguments to the conditional command's =~ operator.
.TP 8
.B dotglob
If set,
.B bash
includes filenames beginning with a `.' in the results of pathname
expansion.
.TP 8
.B execfail
If set, a non-interactive shell will not exit if
it cannot execute the file specified as an argument to the
.B exec
builtin command. An interactive shell does not exit if

.B exec
fails.
.TP 8
.B expand_aliases
If set, aliases are expanded as described above under
.SM
.BR ALIASES .
This option is enabled by default for interactive shells.
.TP 8
.B extdebug
If set, behavior intended for use by debuggers is enabled:
.RS
.TP
.B 1.
The \fB\-F\fP option to the \fBdeclare\fP builtin displays the source
file name and line number corresponding to each function name supplied
as an argument.
.TP
.B 2.
If the command run by the \fBDEBUG\fP trap returns a non-zero value, the
next command is skipped and not executed.
.TP
.B 3.
If the command run by the \fBDEBUG\fP trap returns a value of 2, and the
shell is executing in a subroutine (a shell function or a shell script
executed by the \fB.\fP or \fBsource\fP builtins), a call to
\fBreturn\fP is simulated.
.TP
.B 4.
\fBBASH_ARGC\fP and \fBBASH_ARGV\fP are updated as described in their
descriptions above.
.TP
.B 5.
Function tracing is enabled: command substitution, shell functions, and
subshells invoked with \fB(\fP \fIcommand\fP \fB)\fP inherit the
\fBDEBUG\fP and \fBRETURN\fP traps.
.TP
.B 6.
Error tracing is enabled: command substitution, shell functions, and
subshells invoked with \fB(\fP \fIcommand\fP \fB)\fP inherit the
\fBERROR\fP trap.
.RE
.TP 8
.B extglob
If set, the extended pattern matching features described above under
\fBPathname Expansion\fP are enabled.
.TP 8
.B extquote
If set, \fB$\fP\(aq\fIstring\fP\(aq and \fB$\fP"\fIstring\fP" quoting is
performed within \fB${\fP\fIparameter\fP\fB}\fP expansions
enclosed in double quotes. This option is enabled by default.
.TP 8
.B failglob
If set, patterns which fail to match filenames during pathname expansion
result in an expansion error.
.TP 8
.B force_fignore
If set, the suffixes specified by the \fBFIGNORE\fP shell variable
cause words to be ignored when performing word completion even if
the ignored words are the only possible completions.
See
.SM
\fBSHELL VARIABLES\fP
above for a description of \fBFIGNORE\fP.
This option is enabled by default.
.TP 8
.B gnu_errfmt
If set, shell error messages are written in the standard GNU error
message format.
.TP 8
.B histappend
If set, the history list is appended to the file named by the value
of the
.B HISTFILE
variable when the shell exits, rather than overwriting the file.
.TP 8
.B histreedit
If set, and
.B readline
is being used, a user is given the opportunity to re-edit a
failed history substitution.
.TP 8
.B histverify
If set, and
.B readline
is being used, the results of history substitution are not immediately
passed to the shell parser. Instead, the resulting line is loaded into
the \fBreadline\fP editing buffer, allowing further modification.
.TP 8
.B hostcomplete
If set, and
.B readline
is being used, \fBbash\fP will attempt to perform hostname completion when a
word containing a \fB@\fP is being completed (see
.B Completing
under
.SM
.B READLINE
above).
This is enabled by default.
.TP 8
.B huponexit
If set, \fBbash\fP will send
.SM
.B SIGHUP
to all jobs when an interactive login shell exits.
.TP 8
.B interactive_comments
If set, allow a word beginning with
.B #
to cause that word and all remaining characters on that
line to be ignored in an interactive shell (see
.SM
.B COMMENTS
above). This option is enabled by default.
.TP 8
.B lithist
If set, and the
.B cmdhist
option is enabled, multi-line commands are saved to the history with
embedded newlines rather than using semicolon separators where possible.
.TP 8

.B login_shell
The shell sets this option if it is started as a login shell (see
.SM
.B "INVOCATION"
above).
The value may not be changed.
.TP 8
.B mailwarn
If set, and a file that \fBbash\fP is checking for mail has been
accessed since the last time it was checked, the message ``The mail in
\fImailfile\fP has been read'' is displayed.
.TP 8
.B no_empty_cmd_completion
If set, and
.B readline
is being used,
.B bash
will not attempt to search the \fBPATH\fP for possible completions when
completion is attempted on an empty line.
.TP 8
.B nocaseglob
If set,
.B bash
matches filenames in a case\-insensitive fashion when performing pathname
expansion (see
.B Pathname Expansion
above).
.TP 8
.B nocasematch
If set,
.B bash
matches patterns in a case\-insensitive fashion when performing matching
while executing \fBcase\fP or \fB[[\fP conditional commands.
.TP 8
.B nullglob
If set,
.B bash
allows patterns which match no
files (see
.B Pathname Expansion
above)
to expand to a null string, rather than themselves.
.TP 8
.B progcomp
If set, the programmable completion facilities (see
\fBProgrammable Completion\fP above) are enabled.
This option is enabled by default.
.TP 8
.B promptvars
If set, prompt strings undergo
parameter expansion, command substitution, arithmetic
expansion, and quote removal after being expanded as described in
.SM
.B PROMPTING
above. This option is enabled by default.
.TP 8
.B restricted_shell
The shell sets this option if it is started in restricted mode (see
.SM
.B "RESTRICTED SHELL"
below).
The value may not be changed.
This is not reset when the startup files are executed, allowing
the startup files to discover whether or not a shell is restricted.
.TP 8
.B shift_verbose
If set, the
.B shift
builtin prints an error message when the shift count exceeds the
number of positional parameters.
.TP 8
.B sourcepath
If set, the
\fBsource\fP (\fB.\fP) builtin uses the value of
.SM
.B PATH
to find the directory containing the file supplied as an argument.
This option is enabled by default.
.TP 8
.B xpg_echo
If set, the \fBecho\fP builtin expands backslash-escape sequences
by default.
.RE
.TP
\fBsuspend\fP [\fB\-f\fP]
Suspend the execution of this shell until it receives a
.SM
.B SIGCONT
signal. The
.B \-f
option says not to complain if this is
a login shell; just suspend anyway. The return status is 0 unless
the shell is a login shell and
.B \-f
is not supplied, or if job control is not enabled.
.TP
\fBtest\fP \fIexpr\fP
.PD 0
.TP
\fB[\fP \fIexpr\fP \fB]\fP
Return a status of 0 or 1 depending on
the evaluation of the conditional expression
.IR expr .
Each operator and operand must be a separate argument.
Expressions are composed of the primaries described above under
.SM
.BR "CONDITIONAL EXPRESSIONS" .
\fBtest\fP does not accept any options, nor does it accept and ignore
an argument of \fB\-\-\fP as signifying the end of options.
.if t .sp 0.5
.if n .sp 1
Expressions may be combined using the following operators, listed
in decreasing order of precedence.
.RS
.PD 0
.TP
.B ! \fIexpr\fP
True if
.I expr
is false.
.TP
.B (\fIexpr\fP)

Returns the value of \fIexpr\fP.
This may be used to override the normal precedence of operators.
.TP
\fIexpr1\fP \-\fBa\fP \fIexpr2\fP
True if both
.I expr1
and
.I expr2
are true.
.TP
\fIexpr1\fP \-\fBo\fP \fIexpr2\fP
True if either
.I expr1
or
.I expr2
is true.
.PD
.PP
\fBtest\fP and \fB[\fP evaluate conditional
expressions using a set of rules based on the number of arguments.
.if t .sp 0.5
.if n .sp 1
.PD 0
.TP
0 arguments
The expression is false.
.TP
1 argument
The expression is true if and only if the argument is not null.
.TP
2 arguments
If the first argument is \fB!\fP, the expression is true if and
only if the second argument is null.
If the first argument is one of the unary conditional operators listed above
under
.SM
.BR "CONDITIONAL EXPRESSIONS" ,
the expression is true if the unary test is true.
If the first argument is not a valid unary conditional operator, the expression
is false.
.TP
3 arguments
If the second argument is one of the binary conditional operators listed above
under
.SM
.BR "CONDITIONAL EXPRESSIONS" ,
the result of the expression is the result of the binary test using
the first and third arguments as operands.
If the first argument is \fB!\fP, the value is the negation of
the two-argument test using the second and third arguments.
If the first argument is exactly \fB(\fP and the third argument is
exactly \fB)\fP, the result is the one-argument test of the second
argument.
Otherwise, the expression is false.
The \fB\-a\fP and \fB\-o\fP operators are considered binary operators
in this case.
.TP
4 arguments
If the first argument is \fB!\fP, the result is the negation of
the three-argument expression composed of the remaining arguments.
Otherwise, the expression is parsed and evaluated according to
precedence using the rules listed above.
.TP
5 or more arguments
The expression is parsed and evaluated according to precedence
using the rules listed above.
.RE
.PD
.TP
.B times
Print the accumulated user and system times for the shell and
for processes run from the shell. The return status is 0.
.TP
\fBtrap\fP [\fB\-lp\fP] [[\fIarg\fP] \fIsigspec\fP ...]
The command
.I arg
is to be read and executed when the shell receives
signal(s)
.IR sigspec .
If
.I arg
is absent (and there is a single \fIsigspec\fP) or
.BR \- ,
each specified signal is
reset to its original disposition (the value it had
upon entrance to the shell).
If
.I arg
is the null string the signal specified by each
.I sigspec
is ignored by the shell and by the commands it invokes.
If
.I arg
is not present and
.B \-p
has been supplied, then the trap commands associated with each
.I sigspec
are displayed.
If no arguments are supplied or if only
.B \-p
is given,
.B trap
prints the list of commands associated with each signal.
The
.B \-l
option causes the shell to print a list of signal names and
their corresponding numbers.
Each
.I sigspec
is either
a signal name defined in <\fIsignal.h\fP>, or a signal number.
Signal names are case insensitive and the SIG prefix is optional.
If a
.I sigspec
is
.SM
.B EXIT
(0) the command
.I arg
is executed on exit from the shell.
If a
.I sigspec

```
is
.SM
.BR DEBUG ,
the command
.I arg
is executed before every \fIsimple command\fP, \fIfor\fP command,
\fIcase\fP command, \fIselect\fP command, every arithmetic \fIfor\fP
command, and before the first command executes in a shell function (see
.SM
.B SHELL GRAMMAR
above).
Refer to the description of the \fBextdebug\fP option to the
\fBshopt\fP builtin for details of its effect on the \fBDEBUG\fP trap.
If a
.I sigspec
is
.SM
.BR ERR ,
the command
.I arg
is executed whenever a simple command has a non\-zero exit status,
subject to the following conditions.
The
.SM
.B ERR
trap is not executed if the failed
command is part of the command list immediately following a
.B while
or
.B until
keyword,
part of the test in an
.I if
statement, part of a
.B &&
or
.B \(bv\(bv
list, or if the command's return value is
being inverted via
.BR ! .
These are the same conditions obeyed by the \fBerrexit\fP option.
If a
.I sigspec
is
.SM
.BR RETURN ,
the command
.I arg
is executed each time a shell function or a script executed with the
\fB.\fP or \fBsource\fP builtins finishes executing.
Signals ignored upon entry to the shell cannot be trapped or reset.
Trapped signals that are not being ignored are reset to their original
values in a child process when it is created.
The return status is false if any
.I sigspec
is invalid; otherwise
.B trap
returns true.
.TP
\fBtype\fP [\fB\-aftpP\fP] \fIname\fP [\fIname\fP ...]
With no options,
indicate how each
.I name
would be interpreted if used as a command name.
If the
.B \-t
option is used,
.B type
prints a string which is one of
.IR alias ,
.IR keyword ,
.IR function ,
.IR builtin ,
or
.I file
if
.I name
is an alias, shell reserved word, function, builtin, or disk file,
respectively.
If the
.I name
is not found, then nothing is printed, and an exit status of false
is returned.
If the
.B \-p
option is used,
.B type
either returns the name of the disk file
that would be executed if
.I name
were specified as a command name,
or nothing if
.if t \f(CWtype -t name\fP
.if n ``type -t name''
would not return
.IR file .
The
.B \-P
option forces a
.SM
.B PATH
search for each \fIname\fP, even if
.if t \f(CWtype -t name\fP
.if n ``type -t name''
would not return
.IR file .
If a command is hashed,
.B \-p
and
.B \-P
print the hashed value, not necessarily the file that appears
first in
.SM
.BR PATH .
If the
.B \-a
option is used,
.B type
prints all of the places that contain
an executable named
.IR name .
This includes aliases and functions,
```

if and only if the
.B \-p
option is not also used.
The table of hashed commands is not consulted
when using
.BR \-a .
The
.B \-f
option suppresses shell function lookup, as with the \fBcommand\fP builtin.
.B type
returns true if any of the arguments are found, false if
none are found.
.TP
\fBulimit\fP [\fB\-SHacdefilmnpqrstuvx\fP [\fIlimit\fP]]
Provides control over the resources available to the shell and to
processes started by it, on systems that allow such control.
The \fB\-H\fP and \fB\-S\fP options specify that the hard or soft limit is
set for the given resource. A hard limit cannot be increased once it
is set; a soft limit may be increased up to the value of the hard limit.
If neither \fB\-H\fP nor \fB\-S\fP is specified, both the soft and hard
limits are set.
The value of
.I limit
can be a number in the unit specified for the resource
or one of the special values
.BR hard ,
.BR soft ,
or
.BR unlimited ,
which stand for the current hard limit, the current soft limit, and
no limit, respectively.
If
.I limit
is omitted, the current value of the soft limit of the resource is
printed, unless the \fB\-H\fP option is given. When more than one
resource is specified, the limit name and unit are printed before the value.
Other options are interpreted as follows:
.RS
.PD 0
.TP
.B \-a
All current limits are reported
.TP
.B \-c
The maximum size of core files created
.TP
.B \-d
The maximum size of a process's data segment
.TP
.B \-e
The maximum scheduling priority ("nice")
.TP
.B \-f
The maximum size of files written by the shell and its children
.TP
.B \-i
The maximum number of pending signals
.TP
.B \-l
The maximum size that may be locked into memory
.TP
.B \-m
The maximum resident set size
.TP
.B \-n
The maximum number of open file descriptors (most systems do not
allow this value to be set)
.TP
.B \-p
The pipe size in 512-byte blocks (this may not be set)
.TP
.B \-q
The maximum number of bytes in POSIX message queues
.TP
.B \-r
The maximum real-time scheduling priority
.TP
.B \-s
The maximum stack size
.TP
.B \-t
The maximum amount of cpu time in seconds
.TP
.B \-u
The maximum number of processes available to a single user
.TP
.B \-v
The maximum amount of virtual memory available to the shell
.TP
.B \-x
The maximum number of file locks
.PD
.PP
If
.I limit
is given, it is the new value of the specified resource (the
.B \-a
option is display only).
If no option is given, then
.B \-f
is assumed. Values are in 1024-byte increments, except for
.BR \-t ,
which is in seconds,
.BR \-p ,
which is in units of 512-byte blocks,
and
.B \-n
and
.BR \-u ,
which are unscaled values.
The return status is 0 unless an invalid option or argument is supplied,
or an error occurs while setting a new limit.
.RE
.TP
\fBumask\fP [\fB\-p\fP] [\fB\-S\fP] [\fImode\fP]
The user file-creation mask is set to
.IR mode .
If
.I mode
begins with a digit, it
is interpreted as an octal number; otherwise
it is interpreted as a symbolic mode mask similar

to that accepted by
.IR chmod (1).
If
.I mode
is omitted, the current value of the mask is printed.
The
.B \-S
option causes the mask to be printed in symbolic form; the
default output is an octal number.
If the
.B \-p
option is supplied, and
.I mode
is omitted, the output is in a form that may be reused as input.
The return status is 0 if the mode was successfully changed or if
no \fImode\fP argument was supplied, and false otherwise.
.TP
\fBunalias\fP [\-\fBa\fP] [\fIname\fP ...]
Remove each \fIname\fP from the list of defined aliases. If
.B \-a
is supplied, all alias definitions are removed. The return
value is true unless a supplied
.I name
is not a defined alias.
.TP
\fBunset\fP [\-\fBfv\fP] [\fIname\fP ...]
For each
.IR name ,
remove the corresponding variable or function.
If no options are supplied, or the
.B \-v
option is given, each
.I name
refers to a shell variable.
Read-only variables may not be unset.
If
.B \-f
is specified, each
.I name
refers to a shell function, and the function definition
is removed.
Each unset variable or function is removed from the environment
passed to subsequent commands.
If any of
.SM
.BR RANDOM ,
.SM
.BR SECONDS ,
.SM
.BR LINENO ,
.SM
.BR HISTCMD ,
.SM
.BR FUNCNAME ,
.SM
.BR GROUPS ,
or
.SM
.B DIRSTACK
are unset, they lose their special properties, even if they are
subsequently reset. The exit status is true unless a
.I name
is readonly.
.TP
\fBwait\fP [\fIn ...\fP]
Wait for each specified process and return its termination status.
Each
.I n
may be a process
ID or a job specification; if a job spec is given, all processes
in that job's pipeline are waited for. If
.I n
is not given, all currently active child processes
are waited for, and the return status is zero. If
.I n
specifies a non-existent process or job, the return status is
127. Otherwise, the return status is the exit status of the last
process or job waited for.
.\" bash_builtins
.if \n(zZ=1 .ig zZ
.SH "RESTRICTED SHELL"
.\" rbash.1
.zY
.PP
If
.B bash
is started with the name
.BR rbash ,
or the
.B \-r
option is supplied at invocation,
the shell becomes restricted.
A restricted shell is used to
set up an environment more controlled than the standard shell.
It behaves identically to
.B bash
with the exception that the following are disallowed or not performed:
.IP \(bu
changing directories with \fBcd\fP
.IP \(bu
setting or unsetting the values of
.BR SHELL ,
.BR PATH ,
.BR ENV ,
or
.B BASH_ENV
.IP \(bu
specifying command names containing
.B /
.IP \(bu
specifying a file name containing a
.B /
as an argument to the
.B .
builtin command
.IP \(bu
Specifying a filename containing a slash as an argument to the
.B \-p
option to the
.B hash
builtin command
.IP \(bu

importing function definitions from the shell environment at startup
.IP \(bu
parsing the value of \fBSHELLOPTS\fP from the shell environment at startup
.IP \(bu
redirecting output using the >, >|, <>, >&, &>, and >> redirection operators
.IP \(bu
using the
.B exec
builtin command to replace the shell with another command
.IP \(bu
adding or deleting builtin commands with the
.B \-f
and
.B \-d
options to the
.B enable
builtin command
.IP \(bu
Using the \fBenable\fP builtin command to enable disabled shell builtins
.IP \(bu
specifying the
.B \-p
option to the
.B command
builtin command
.IP \(bu
turning off restricted mode with
\fBset +r\fP or \fBset +o restricted\fP.
.PP
These restrictions are enforced after any startup files are read.
.PP
.ie \n(zY=1 When a command that is found to be a shell script is executed,
.el \{ When a command that is found to be a shell script is executed
(see
.SM
.B "COMMAND EXECUTION"
above),
\}
.B rbash
turns off any restrictions in the shell spawned to execute the
script.
.\" end of rbash.1
.if \n(zY=1 .ig zY
.SH "SEE ALSO"
.PD 0
.TP
\fIBash Reference Manual\fP, Brian Fox and Chet Ramey
.TP
\fIThe Gnu Readline Library\fP, Brian Fox and Chet Ramey
.TP
\fIThe Gnu History Library\fP, Brian Fox and Chet Ramey
.TP
\fIPortable Operating System Interface (POSIX) Part 2: Shell and Utilities\fP, IEEE
.TP
\fIsh\fP(1), \fIksh\fP(1), \fIcsh\fP(1)
.TP
\fIemacs\fP(1), \fIvi\fP(1)
.TP
\fIreadline\fP(3)
.PD
.SH FILES
.PD 0
.TP
.FN /bin/bash
The \fBbash\fP executable
.TP
.FN /etc/profile
The systemwide initialization file, executed for login shells
.TP
.FN ~/.bash_profile
The personal initialization file, executed for login shells
.TP
.FN ~/.bashrc
The individual per-interactive-shell startup file
.TP
.FN ~/.bash_logout
The individual login shell cleanup file, executed when a login shell exits
.TP
.FN ~/.inputrc
Individual \fIreadline\fP initialization file
.PD
.SH AUTHORS
Brian Fox, Free Software Foundation
.br
bfox@gnu.org
.PP
Chet Ramey, Case Western Reserve University
.br
chet@po.cwru.edu
.SH BUG REPORTS
If you find a bug in
.B bash,
you should report it. But first, you should
make sure that it really is a bug, and that it appears in the latest
version of
.BR bash .
The latest version is always available from
\fIftp://ftp.gnu.org/pub/bash/\fP.
.PP
Once you have determined that a bug actually exists, use the
.I bashbug
command to submit a bug report.
If you have a fix, you are encouraged to mail that as well!
Suggestions and `philosophical' bug reports may be mailed
to \fIbug-bash@gnu.org\fP or posted to the Usenet
newsgroup
.BR gnu.bash.bug .
.PP
ALL bug reports should include:
.PP
.PD 0
.TP 20
The version number of \fBbash\fR
.TP
The hardware and operating system
.TP
The compiler used to compile
.TP
A description of the bug behaviour
.TP
A short script or `recipe' which exercises the bug
.PD

```
.PP
.I bashbug
inserts the first three items automatically into the template
it provides for filing a bug report.
.PP
Comments and bug reports concerning
this manual page should be directed to
.IR chet@po.cwru.edu .
.SH BUGS
.PP
It's too big and too slow.
.PP
There are some subtle differences between
.B bash
and traditional versions of
.BR sh ,
mostly because of the
.SM
.B POSIX
specification.
.PP
Aliases are confusing in some uses.
.PP
Shell builtin commands and functions are not stoppable/restartable.
.PP
Compound commands and command sequences of the form 'a ; b ; c'
are not handled gracefully when process suspension is attempted.
When a process is stopped, the shell immediately executes the next
command in the sequence.
It suffices to place the sequence of commands between
parentheses to force it into a subshell, which may be stopped as
a unit.
.PP
Commands inside of \fB$(\fP...\fB)\fP command substitution are not
parsed until substitution is attempted.  This will delay error
reporting until some time after the command is entered.  For example,
unmatched parentheses, even inside shell comments, will result in
error messages while the construct is being read.
.PP
Array variables may not (yet) be exported.
.zZ
.zY
[wansongHome@wansong:/usr/share/man/man1]$ ls
```

```
AppleFileServer.1        eyapp5.18.1           mklocale.1                          perldsc5.16.1          perlsub5.18.1         serialver.1
CA.pl.1ssl               false.1               mktemp.1                            perldsc5.18.1          perlsymbian.1         servertool.1
GoogleSyncMigrate.1      fax.1                 mmroff.1                            perldtrace.1           perlsymbian5.16.1     sess_id.1ssl
KeychainMigrator.1       fc.1                  mnthome.1                           perldtrace5.16.1       perlsymbian5.18.1     set.1
WirelessRadioManagerd.1  fddist.1m             more.1.gz                           perldtrace5.18.1       perlsyn.1             setenv.1
'[.1'                    fg.1                  mpioutil.1                          perlebcdic.1           perlsyn5.16.1         setquota.1
a2p.1                    fgrep.1               mptcp_client.1                      perlebcdic5.16.1       perlsyn5.18.1         settc.1
a2p.1.gz                 fi.1                  msgs.1                              perlebcdic5.18.1       perlthanks.1          setty.1
a2p5.16.1                file.1                mv.1                                perlembed.1            perlthanks5.16.1      setuids.d.1m
a2p5.18.1                filebyproc.d.1m       nano.1                              perlembed5.16.1        perlthanks5.18.1      setvar.1
ab.1                     filetest.1            native2ascii.1                      perlembed5.18.1        perlthrtut.1          sftp.1
accesstool.1             filtercalltree.1      nc.1                                perlepoc.1             perlthrtut5.16.1      sh.1
addftinfo.1              find.1                ncal.1                              perlepoc5.16.1         perlthrtut5.18.1      shar.1
afconvert.1              find2perl.1           ncctl.1                             perlexperiment.1       perltie.1             sharing.1
afhash.1                 find2perl5.16.1       ncdestroy.1                         perlexperiment5.16.1   perltie5.16.1         shasum.1
afida.1                  find2perl5.18.1       ncinit.1                            perlexperiment5.18.1   perltie5.18.1         shasum5.16.1
afinfo.1                 findrule.1            nclist.1                            perlfaq.1              perltoc.1             shasum5.18.1
afmtodit.1               findrule5.16.1        ncurses5.4-config.1                 perlfaq1.1             perltoc5.16.1         shift.1
afplay.1                 findrule5.18.1        neqn.1                              perlfaq15.16.1         perltoc5.18.1         shlock.1
afscexpand.1             finger.1              net-server.1                        perlfaq15.18.1         perltodo.1            sigdist.d.1m
agentxtrap.1             fixproc.1             net-server5.16.1                    perlfaq2.1             perltodo5.16.1        sips.1
alias.1                  fmt.1                 net-server5.18.1                    perlfaq25.16.1         perltodo5.18.1        sleep.1
alloc.1                  fold.1                net-snmp-config.1                   perlfaq25.18.1         perltooc.1            slogin.1
applesingle.1            footprint.1           net-snmp-create-v3-user.1           perlfaq3.1             perltooc5.16.1        smbutil.1
appletviewer.1           for.1                 netlog.1                            perlfaq35.16.1         perltooc5.18.1        smcDiagnose.1
apply.1                  foreach.1             netstat.1                           perlfaq35.18.1         perltoot.1            smime.1ssl
apropos.1                from.1                nettop.1                            perlfaq4.1             perltoot5.16.1        sndiskmove.1
apt.1                    fs_usage.1            newaliases.1                        perlfaq45.16.1         perltoot5.18.1        snfsdefrag.1
apxs.1                   ftp.1                 newgrp.1                            perlfaq45.18.1         perltrap.1            snmp-bridge-mib.1
arch.1                   funzip.1              newproc.d.1m                        perlfaq5.1             perltrap5.16.1        snmpbulkget.1
asctl.1                  fuser.1               nfsstat.1                           perlfaq55.16.1         perltrap5.18.1        snmpbulkwalk.1
asn1parse.1ssl           fwkdp.1               nice.1                              perlfaq5.18.1          perltru64.1           snmpcmd.1
assetutil.1              fwkpfv.1              nl.1                                perlfaq55.16.1         perltru645.16.1       snmpconf.1
at.1                     gdiffmk.1             nohup.1                             perlfaq55.18.1         perltru645.18.1       snmpdelta.1
atos.1                   gem.1                 notify.1                            perlfaq6.1             perltw.1              snmpdf.1
atq.1                    gen_bridge_metadata.1 notifyutil.1                        perlfaq65.16.1         perltw5.16.1          snmpget.1
atrm.1                   gencat.1              nroff.1                             perlfaq65.18.1         perltw5.18.1          snmpgetnext.1
auditreduce.1            gendsa.1ssl           nseq.1ssl                           perlfaq7.1             perlunicode.1         snmpinform.1
authopen.1               genrsa.1ssl           nslookup.1                          perlfaq75.16.1         perlunicode5.16.1     snmpnetstat.1
automator.1              getconf.1             nsupdate.1                          perlfaq75.18.1         perlunicode5.18.1     snmpset.1
auval.1                  getopt.1              ntp-keygen.1                        perlfaq8.1             perlunifaq.1          snmpstatus.1
auvaltool.1              getopts.1             ntpd.1                              perlfaq85.16.1         perlunifaq5.16.1      snmptable.1
avbdevice1.1             glob.1                ntpdc.1                             perlfaq85.18.1         perlunifaq5.18.1      snmptest.1
avbdiagnose.1            gnuattach.1           ntpq.1                              perlfaq9.1             perluniintro.1        snmptranslate.1
avbutil.1                gnuclient.1           ocsp.1ssl                           perlfaq95.16.1         perluniintro5.16.1    snmptrap.1
avconvert.1              gnudoit.1             ocspd.1                             perlfaq95.18.1         perluniintro5.18.1    snmpusm.1
awk.1                    gnuserv.1             od.1                                perlfilter.1           perluniprops.1        snmpvacm.1
axreg.1                  goto.1                odmigrationtool.1                   perlfilter5.16.1       perluniprops5.16.1    snmpwalk.1
base64.1                 grap2graph.1          odutil.1                            perlfilter5.18.1       perluniprops5.18.1    sntp.1
basename.1               graphicssession.1     onintr.1                            perlfork.1             perlunitut.1          soelim.1
bash.1                   grep.1                open.1                              perlfork5.16.1         perlunitut5.16.1      sort.1
bashbug.1                grn.1                 opensnoop.1m                        perlfork5.18.1         perlunitut5.18.1      source.1
batch.1                  grodvi.1              openssl.1ssl                        perlform.1             perlutil.1            speed.1ssl
bc.1                     groff.1               orbd.1                              perlform5.16.1         perlutil5.16.1        spfd.1
bg.1                     groffer.1             osacompile.1                        perlform5.18.1         perlutil5.18.1        spfd5.16.1
biff.1                   grog.1                osadecompile.1                      perlfreebsd.1          perluts.1             spfd5.18.1
bind.1                   grohtml.1             osalang.1                           perlfreebsd5.16.1      perluts5.16.1         spfquery.1
bindkey.1                grolbp.1              osascript.1                         perlfreebsd5.18.1      perluts5.18.1         spfquery5.16.1
binhex.1                 grolj4.1              ostraceutil.1                       perlfunc.1             perlvar.1             spfquery5.18.1
binhex.pl.1              grops.1               pack200.1                           perlfunc5.16.1         perlvar5.16.1         spkac.1ssl
binhex.pl5.18.1          grotty.1              package-stash-conflicts.1           perlfunc5.18.1         perlvar5.18.1         splain.1
bitesize.d.1m            groups.1              package-stash-conflicts5.18.1       perlgit.1             perlvmesa.1           splain5.16.1
brctl.1                  gunzip.1              pagesize.1                          perlgit5.16.1          perlvmesa5.16.1       splain5.18.1
break.1                  gzcat.1               par.p1.1                            perlgit5.18.1          perlvms.1             split.1
breaksw.1                gzexe.1               par.pl5.16.1                        perlglossary.1         perlvms5.16.1         srm.1
bsdtar.1                 gzip.1                par.pl5.18.1                        perlglossary5.16.1     perlvms5.18.1         ssh-add.1
bspatch.1                h2ph.1                parl.1                              perlglossary5.18.1     perlvos.1             ssh-agent.1
builtin.1                h2ph5.16.1            parl5.16.1                          perlgpl.1             perlvos5.16.1         ssh-keygen.1
builtins.1               h2ph5.18.1            parl5.18.1                          perlgpl5.16.1          perlvos5.18.1         ssh-keyscan.1
bunzip2.1                h2xs.1                passwd.1                            perlgpl5.18.1          perlwin32.1           ssh.1
bzcat.1                  h2xs5.16.1            passwd.1ssl                         perlguts.1             perlwin325.16.1       stat.1
bzcmp.1                  h2xs5.18.1            paste.1                             perlguts5.16.1         perlwin325.18.1       stop.1
bzdiff.1                 hash.1                patch.1                             perlguts5.18.1         perlxs.1              stringdups.1
bzegrep.1                hashstat.1            pathchk.1                           perlhack.1             perlxs5.16.1          stringdups32.1
bzfgrep.1                hdiutil.1             pathopens.d.1m                      perlhack5.16.1         perlxs5.18.1          stty.1
bzgrep.1                 head.1                pax.1                               perlhack5.18.1         perlxstut5.16.1       su.1
bzip2.1                  heap.1                pbcopy.1                            perlhacktips.1         perlxstut5.18.1       sum.1
bzip2recover.1           heap32.1              pbpaste.1                           perlhacktips5.16.1     perlxstypemap.1       suspend.1
bzless.1                 hexdump.1             pcap-config.1                       perlhacktips5.18.1     perlxstypemap5.16.1   sw_vers.1
```

bzmore.1
c++filt.1
c2ph.1
c2ph5.16.1
c2ph5.18.1
c_rehash.1ssl
ca.1ssl
cal.1
calendar.1
cancel.1.gz
cap_mkdb.1
captoinfo.1m
case.1
cat.1
cc_fips_test.1
cd.1
cdcontexttool.1
cdinteracttool.1
cdknowledgetool.1
certtool.1
cfilutil.1
chdir.1
checkLocalKDC.1
checknr.1
chflags.1
chfn.1
chgrp.1
chmod.1
chpass.1
chsh.1
ciphers.1ssl
cksum.1
clear.1
cmp.1
codesign.1
col.1
colcrt.1
colldef.1
colrm.1
column.1
comm.1
command.1
complete.1
compress.1
config_data.1
config_data5.16.1
config_data5.18.1
configureLocalKDC.1
continue.1
corelist.1
corelist5.16.1
corelist5.18.1
cp.1
cpan.1
cpan2dist.1
cpan2dist5.16.1
cpan2dist5.18.1
cpan5.16.1
cpan5.18.1
cpanp.1
cpanp5.16.1
cpanp5.18.1
cpio.1
cpuwalk.d.1m
creatbyproc.d.1m
createhomedir.1
crl.1ssl
crl2pkcs7.1ssl
crlrefresh.1
crontab.1
csdiagnose.1
csgather.1
csh.1
csplit.1
csreq.1
cu.1
cups-config.1.gz
cups.1.gz
cupstestdsc.1.gz
cupstestppd.1.gz
curl-config.1
curl.1
cut.1
cvaffinity.1
cvcp.1
cvmkdir.1
cvmkfile.1
cvversions.1
dappprof.1m
dapptrace.1m
darwinup.1
date.1
db_archive.1
db_checkpoint.1
db_deadlock.1
db_dump.1
db_load.1
db_printlog.1
db_recover.1
db_stat.1
db_upgrade.1
db_verify.1
dbicadmin.1
dbicadmin5.16.1
dbicadmin5.18.1
dbilogstrip.1
dbilogstrip5.16.1
dbilogstrip5.18.1
dbiprof.1
dbiprof5.16.1
dbiprof5.18.1
dbiproxy.1
dbiproxy5.16.1
dbiproxy5.18.1
dbmmanage.1
dc.1
dd.1
debinhex.pl.1
debinhex.pl5.18.1
default.1
defaults.1
df.1

history.1
hiutil.1
host.1
hostname.1
hotspot.d.1m
hpftodit.1
htdbm.1
htdigest.1
htpasswd.1
httpdstat.d.1m
httxt2dbm.1
hup.1
iconservicesagent.1
iconservicesd.1
iconset.1
icontool.1
iconutil.1
iconv.1
id.1
idlj.1
if.1
imptrace.1
indxbib.1
info.1
infocmp.1m
infokey.1
infotocap.1m
install-info.1
install.1
instmodsh.1
instmodsh5.16.1
instmodsh5.18.1
intro.1
iofile.d.1m
iofileb.d.1m
iopattern.1m
iopending.1m
iosnoop.1m
iotop.1m
ip2cc.1
ip2cc5.16.1
ip2cc5.18.1
ipcrm.1
ipcs.1
ipmitool.1
ippfind.1.gz
ipptool.1.gz
irb.1
jar.1
jarsigner.1
java.1
java_home.1
javac.1
javadoc.1
javah.1
javap.1
javaws.1
jconsole.1
jdb.1
jhat.1
jinfo.1
jmap.1
jobid.1
jobs.1
join.1
jot.1
jps.1
jrunscript.1
jsadebugd.1
json_pp.1
json_pp5.16.1
json_pp5.18.1
jstack.1
jstat.1
jstatd.1
kcc.1
kdestroy.1
keytool.1
kgetcred.1
kill.1
kill.d.1m
killall.1
kinit.1
klist.1
kpasswd.1
kpconfig.1
kpfile.1
kptop.1
krb5-config.1
ksh.1
kswitch.1
lam.1
languagesetup.1
last.1
lastcomm.1
lastwords.1m
latency.1
launchctl.1
launchservicesd.1
layerutil.1
ldapadd.1
ldapcompare.1
ldapdelete.1
ldapexop.1
ldapmodify.1
ldapmodrdn.1
ldappasswd.1
ldapsearch.1
ldapurl.1
ldapwhoami.1
leaks.1
leaks32.1
leave.1
less.1.gz
lessecho.1.gz
libnetcfg.1
libnetcfg5.16.1
libnetcfg5.18.1
limit.1
link.1
lkbib.1
ln.1

pcsstatus.1
perl.1
perl.1.gz
perl5.16.1
perl5.16.3.1
perl5.16.35.16.1
perl5.165.16.1
perl5.18.1
perl5.18.2.1
perl5.18.25.18.1
perl5.185.18.1
perl5004delta.1
perl5004delta5.16.1
perl5004delta5.18.1
perl5005delta.1
perl5005delta5.16.1
perl5005delta5.18.1
perl5100delta.1
perl5100delta5.16.1
perl5100delta5.18.1
perl5101delta.1
perl5101delta5.16.1
perl5101delta5.18.1
perl5120delta.1
perl5120delta5.16.1
perl5120delta5.18.1
perl5121delta.1
perl5121delta5.16.1
perl5121delta5.18.1
perl5122delta.1
perl5122delta5.16.1
perl5122delta5.18.1
perl5123delta.1
perl5123delta5.16.1
perl5123delta5.18.1
perl5124delta.1
perl5124delta5.16.1
perl5124delta5.18.1
perl5125delta.1
perl5125delta5.16.1
perl5125delta5.18.1
perl5140delta.1
perl5140delta5.16.1
perl5140delta5.18.1
perl5141delta.1
perl5141delta5.16.1
perl5141delta5.18.1
perl5142delta.1
perl5142delta5.16.1
perl5142delta5.18.1
perl5143delta.1
perl5143delta5.16.1
perl5143delta5.18.1
perl5144delta.1
perl5144delta5.16.1
perl5144delta5.18.1
perl5160delta.1
perl5160delta5.16.1
perl5160delta5.18.1
perl5161delta.1
perl5161delta5.16.1
perl5161delta5.18.1
perl5162delta.1
perl5162delta5.16.1
perl5162delta5.18.1
perl5163delta.1
perl5163delta5.16.1
perl5163delta5.18.1
perl5180delta.1
perl5180delta5.18.1
perl5181delta.1
perl5181delta5.18.1
perl5182delta.1
perl5182delta5.18.1
perl561delta.1
perl561delta5.16.1
perl561delta5.18.1
perl56delta.1
perl56delta5.16.1
perl56delta5.18.1
perl581delta.1
perl581delta5.16.1
perl581delta5.18.1
perl582delta.1
perl582delta5.16.1
perl582delta5.18.1
perl583delta.1
perl583delta5.16.1
perl583delta5.18.1
perl584delta.1
perl584delta5.16.1
perl584delta5.18.1
perl585delta.1
perl585delta5.16.1
perl585delta5.18.1
perl586delta.1
perl586delta5.16.1
perl586delta5.18.1
perl587delta.1
perl587delta5.16.1
perl587delta5.18.1
perl588delta.1
perl588delta5.16.1
perl588delta5.18.1
perl589delta.1
perl589delta5.16.1
perl589delta5.18.1
perl58delta.1
perl58delta5.16.1
perl58delta5.18.1
perlaix.1
perlaix5.18.1
perlamiga.1
perlamiga5.16.1
perlamiga5.18.1
perlapi.1
perlapi5.16.1
perlapi5.18.1
perlapio.1
perlapio5.16.1
perlapio5.18.1
perlartistic.1
perlartistic5.16.1

perlhacktut.1
perlhacktut5.16.1
perlhacktut5.18.1
perlhaiku.1
perlhaiku5.16.1
perlhaiku5.18.1
perlhist.1
perlhist5.16.1
perlhist5.18.1
perlhpux.1
perlhpux5.16.1
perlhpux5.18.1
perlhurd.1
perlhurd5.16.1
perlhurd5.18.1
perlintern.1
perlintern5.16.1
perlintern5.18.1
perlinterp.1
perlinterp5.16.1
perlinterp5.18.1
perlintro.1
perlintro5.16.1
perlintro5.18.1
perliol.1
perliol5.16.1
perliol5.18.1
perlipc.1
perlipc5.16.1
perlipc5.18.1
perlirix.1
perlirix5.18.1
perlivp.1
perlivp5.16.1
perlivp5.18.1
perljp.1
perljp5.16.1
perljp5.18.1
perlko.1
perlko5.16.1
perlko5.18.1
perllexwarn.1
perllexwarn5.16.1
perllexwarn5.18.1
perllinux.1
perllinux5.16.1
perllinux5.18.1
perllocale.1
perllocale5.16.1
perllocale5.18.1
perllol.1
perllol5.16.1
perllol5.18.1
perlmacos.1
perlmacos5.16.1
perlmacos5.18.1
perlmacosx.1
perlmacosx5.16.1
perlmacosx5.18.1
perlmod.1
perlmod5.16.1
perlmod5.18.1
perlmodinstall.1
perlmodinstall5.16.1
perlmodinstall5.18.1
perlmodlib.1
perlmodlib5.16.1
perlmodlib5.18.1
perlmodstyle.1
perlmodstyle5.16.1
perlmodstyle5.18.1
perlmpeix.1
perlmpeix5.16.1
perlmroapi.1
perlmroapi5.16.1
perlmroapi5.18.1
perlnetware.1
perlnetware5.16.1
perlnetware5.18.1
perlnewmod.1
perlnewmod5.16.1
perlnewmod5.18.1
perlnumber.1
perlnumber5.16.1
perlnumber5.18.1
perlobj.1
perlobj5.16.1
perlobj5.18.1
perlootut.1
perlootut5.16.1
perlootut5.18.1
perlop.1
perlop5.16.1
perlop5.18.1
perlopenbsd.1
perlopenbsd5.16.1
perlopenbsd5.18.1
perlopentut.1
perlopentut5.16.1
perlopentut5.18.1
perlos2.1
perlos25.16.1
perlos25.18.1
perlos390.1
perlos3905.16.1
perlos3905.18.1
perlos400.1
perlos4005.16.1
perlos4005.18.1
perlpacktut.1
perlpacktut5.16.1
perlpacktut5.18.1
perlperf.1
perlperf5.16.1
perlperf5.18.1
perlplan9.1
perlplan95.16.1
perlplan95.18.1
perlpod.1
perlpod5.16.1
perlpod5.18.1

perlxstypemap5.18.1
pfbtops.1
pgrep.1
phar.1
phar.phar.1
php-config.1
php.1
phpize.1
pic.1
pic2graph.1
pico.1
piconv.1
piconv5.16.1
piconv5.18.1
pidpersec.d.1m
pkcs12.1ssl
pkcs7.1ssl
pkcs8.1ssl
pkgbuild.1
pkgutil.1
pkill.1
pl.1
pl2pm.1
pl2pm5.16.1
pl2pm5.18.1
plockstat.1
plutil.1
pmset.1
pod2html.1
pod2html5.16.1
pod2html5.18.1
pod2latex.1
pod2latex5.16.1
pod2latex5.18.1
pod2man.1
pod2man5.16.1
pod2man5.18.1
pod2readme.1
pod2readme5.16.1
pod2readme5.18.1
pod2text.1
pod2text5.16.1
pod2text5.18.1
pod2usage.1
pod2usage5.16.1
pod2usage5.18.1
podchecker.1
podchecker5.16.1
podchecker5.18.1
podselect.1
podselect5.16.1
podselect5.18.1
policytool.1
popd.1
post-grohtml.1
postalias.1
postcat.1
postconf.1
postdrop.1
postfix.1
postkick.1
postlock.1
postlog.1
postmap.1
postmulti.1
postqueue.1
postsuper.1
powermetrics.1
pp.1
pp5.16.1
pp5.18.1
ppdc.1.gz
ppdhtml.1.gz
ppdi.1.gz
ppdmerge.1.gz
ppdpo.1.gz
pr.1
praudit.1
pre-grohtml.1
priclass.d.1m
pridist.d.1m
printenv.1
printf.1
procsystime.1m
productbuild.1
productsign.1
profiles.1
prove.1
prove5.16.1
prove5.18.1
ps.1
psed.1
psed5.16.1
psed5.18.1
pstopdf.1
pstruct.1
pstruct5.16.1
pstruct5.18.1
ptar.1
ptar5.16.1
ptar5.18.1
ptardiff.1
ptardiff5.16.1
ptardiff5.18.1
ptargrep.1
ptargrep5.16.1
ptargrep5.18.1
pubsub.1
pushd.1
pwd.1
pwhich.1
pwhich5.16.1
pwhich5.18.1
pydoc.1
pydoc2.6.1
pydoc2.7.1
python.1
python2.6.1
python2.7.1
pythonw.1
pythonw2.6.1
pythonw2.7.1

switch.1
symbols.1.gz
symbolscache.1.gz
syscallbypid.d.1m
syscallbyproc.d.1m
syscallbysysc.d.1m
sysdiagnose.1
syslog.1
systemsoundserverd.1
tab2space.1
tabs.1
tail.1
talk.1
tar.1
taskinfo.1
tbl.1
tccutil.1
tclsh.1tcl
tclsh8.4.1tcl
tclsh8.5.1tcl
tcpdump.1
tcsh.1
tee.1
telltc.1
telnet.1
test.1
texi2dvi.1
texindex.1
tfmtodit.1
tftp.1
then.1
tic.1m
tidy.1
tiff2icns.1
tiffutil.1
time.1
timerfires.1
times.1
tkcon.1
tkmib.1
tkpp.1
tkpp5.16.1
tkpp5.18.1
tnameserv.1
toe.1m
top.1
tops.1
topsyscall.1m
topsysproc.1m
touch.1
tput.1
tr.1
trace.1
trap.1
traptoemail.1
treereg.1
treereg5.16.1
treereg5.18.1
troff.1
true.1
tset.1
tsort.1
tty.1
type.1
ul.1
ulimit.1
umask.1
unalias.1
uname.1
uncomplete.1
uncompress.1
unexpand.1
unhash.1
uniq.1
units.1
unlimit.1
unlink.1
unpack200.1
unset.1
unsetenv.1
until.1
unvis.1
unzip.1
unzipsfx.1
update_dyld_shared_cache.1
uptime.1
users.1
uuconv.1
uucp.1
uudecode.1
uuencode.1
uuidgen.1
uulog.1
uuname.1
uupick.1
uustat.1
uuto.1
uux.1
verify.1ssl
version.1ssl
vi.1
view.1
vim.1
vimdiff.1
vimtutor.1
vis.1
vm_stat.1
vmmap.1
vmmap32.1
w.1
wait.1
wait4path.1
wall.1
wc.1
weblatency.d.1m
what.1
whatis.1
where.1
whereis.1
which.1
while.1

```
dhparam.1ssl          loads.d.1m             perlartistic5.18.1     perlpodspec.1          qc2movie.1            who.1
diff.1                locale.1               perlbeos.1             perlpodspec5.16.1      qlmanage.1            whoami.1
diff3.1               localedef.1            perlbeos5.16.1         perlpodspec5.18.1      qtdefaults.1          whois.1
diffstat.1            locate.1               perlbook.1             perlpodstyle.1         qtmodernizer.1        wish.1tcl
dig.1                 locationCLI.1          perlbook5.16.1         perlpodstyle5.16.1     quota.1               wish8.4.1tcl
dirname.1             lockstat.1             perlbook5.18.1         perlpodstyle5.18.1     rake.1                wish8.5.1tcl
dirs.1                log.1                  perlboot.1             perlpolicy.1           rand.1ssl             write.1
diskhits.1m           logger.1               perlboot5.16.1         perlpolicy5.16.1       rcp.1                 wsgen.1
dispqlen.d.1m         login.1                perlboot5.18.1         perlpolicy5.18.1       read.1                wsimport.1
ditto.1               logname.1              perlbot.1              perlport.1             readlink.1            x509.1ssl
dns-sd.1              logout.1               perlbot5.16.1          perlport5.16.1         readonly.1            xar.1
do.1                  logresolve.1           perlbot5.18.1          perlport5.18.1         red.1                 xargs.1
domainname.1          look.1                 perlbs2000.1           perlpragma.1           refer.1               xattr.1
done.1                lookbib.1              perlbs20005.16.1       perlpragma5.16.1       rehash.1              xcode-select.1
dot_clean.1           lookupd.1              perlbs20005.18.1       perlpragma5.18.1       repeat.1              xgettext.pl.1
drutil.1              lp.1.gz                perlbug.1              perlqnx.1              req.1ssl              xgettext.pl5.16.1
dsa.1ssl              lpoptions.1.gz         perlbug5.16.1          perlqnx5.16.1          reset.1               xgettext.pl5.18.1
dsaparam.1ssl         lpq.1.gz               perlbug5.18.1          perlqnx5.18.1          return.1              xip.1
dscacheutil.1         lpr.1.gz               perlcall.1             perlre.1               rev.1                 xjc.1
dscl.1                lprm.1.gz              perlcall5.16.1         perlre5.16.1           ri.1                  xml2-config.1
dsconfigldap.1        lpstat.1.gz            perlcall5.18.1         perlre5.18.1           rlogin.1              xmlcatalog.1
dsexport.1            ls-F.1                 perlce.1               perlreapi.1            rm.1                  xmllint.1
dsimport.1            ls.1                   perlce5.16.1           perlreapi5.16.1        rmdir.1               xsltproc.1
dsmemberutil.1        lskq.1                 perlce5.18.1           perlreapi5.18.1        rmic.1                xsubpp.1
dtrace.1              lsm.1                  perlcheat.1            perlrebackslash.1      rmid.1                xsubpp5.16.1
dtruss.1m             lsvfs.1                perlcheat5.16.1        perlrebackslash5.16.1  rmiregistry.1         xsubpp5.18.1
du.1                  lwp-download.1         perlcheat5.18.1        perlrebackslash5.18.1  rs.1                  xxd.1
dyld.1                lwp-download5.16.1     perlclib.1             perlrecharclass.1      rsa.1ssl              yes.1
ec.1ssl               lwp-download5.18.1     perlclib5.16.1         perlrecharclass5.16.1  rsautl.1ssl           ypcat.1
echo.1                lwp-dump.1             perlclib5.18.1         perlrecharclass5.18.1  rsh.1                 ypmatch.1
echotc.1              lwp-dump5.16.1         perlcn.1               perlref.1              rsync.1.gz            ypwhich.1
ecparam.1ssl          lwp-dump5.18.1         perlcn5.16.1           perlref5.16.1          ruby.1                zcat.1
ed.1                  lwp-mirror.1           perlcn5.18.1           perlref5.18.1          runocc.d.1m           zcmp.1
efax.1                lwp-mirror5.16.1       perlcommunity.1        perlreftut.1           ruptime.1             zdiff.1
efix.1                lwp-mirror5.18.1       perlcommunity5.16.1    perlreftut5.16.1       rview.1               zegrep.1
egrep.1               lwp-request.1          perlcommunity5.18.1    perlreftut5.18.1       rvim.1                zfgrep.1
elif.1                lwp-request5.16.1      perlcygwin.1           perlreguts.1           rwbypid.d.1m          zforce.1
else.1                lwp-request5.18.1      perlcygwin5.16.1       perlreguts5.16.1       rwbytype.d.1m         zgrep.1
emacs-undumped.1      macbinary.1            perlcygwin5.18.1       perlreguts5.18.1       rwho.1                zip.1
emacs.1               macerror.1             perldata.1             perlrepository.1       rwsnoop.1m            zipcloak.1
emacsclient.1         macerror5.16.1         perldata5.16.1         perlrepository5.18.1   s2p.1                 zipdetails.1
enc.1ssl              macerror5.18.1         perldata5.18.1         perlrequick.1          s2p5.16.1             zipdetails5.16.1
enc2xs.1              machine.1              perldbmfilter.1        perlrequick5.16.1      s2p5.18.1             zipdetails5.18.1
enc2xs5.16.1          mail.1                 perldbmfilter5.16.1    perlrequick5.18.1      s_client.1ssl         zipgrep.1
enc2xs5.18.1          mailq.1                perldbmfilter5.18.1    perlreref.1            s_server.1ssl         zipinfo.1
encode_keychange.1    mailx.1                perldebguts.1          perlreref5.16.1        s_time.1ssl           zipnote.1
end.1                 makeinfo.1             perldebguts5.16.1      perlreref5.18.1        safaridriver.1        zipsplit.1
endif.1               malloc_history.1       perldebguts5.18.1      perlretut.1            sample.1              zless.1
endsw.1               malloc_history32.1     perldebtut.1           perlretut5.16.1        sampleproc.1m         zmore.1
env.1                 man.1                  perldebtut5.16.1       perlretut5.18.1        sandbox-exec.1        znew.1
eqn.1                 manpath.1              perldebtut5.18.1       perlriscos.1           sar.1                 zprint.1
eqn2graph.1           matchTool.1            perldebug.1            perlriscos5.16.1       say.1                 zsh.1
erb.1                 mcxalr.1               perldebug5.16.1        perlriscos5.18.1       sc_usage.1            zshall.1
errinfo.1m            mcxquery.1             perldebug5.18.1        perlrun.1              scandeps.pl.1         zshbuiltins.1
errstr.1ssl           mcxrefresh.1           perldelta.1            perlrun5.16.1          scandeps.pl5.16.1     zshcompctl.1
esac.1                md5.1                  perldelta5.16.1        perlrun5.18.1          scandeps.pl5.18.1     zshcompsys.1
etags.1               mdfind.1               perldelta5.18.1        perlsec.1              sched.1               zshcompwid.1
eval.1                mdimport.1             perldgux.1             perlsec5.16.1          schemagen.1           zshcontrib.1
evim.1                mdimport32.1           perldgux5.16.1         perlsec5.18.1          scp.1                 zshexpn.1
ex.1                  mdls.1                 perldgux5.18.1         perlsolaris.1          screen.1              zshmisc.1
exec.1                mdmclient.1            perldiag.1             perlsolaris5.16.1      screencapture.1       zshmodules.1
execsnoop.1m          mdutil.1               perldiag5.16.1         perlsolaris5.18.1      script.1              zshoptions.1
exit.1                memberd.1              perldiag5.18.1         perlsource.1           sdiff.1               zshparam.1
expand.1              memory_pressure.1      perldoc.1              perlsource5.16.1       sdx.1                 zshroadmap.1
expect.1              mesg.1                 perldoc5.16.1          perlsource5.18.1       security.1            zshtcpsys.1
export.1              mib2c-update.1         perldoc5.18.1          perlstyle.1            securityd.1           zshzftpsys.1
expr.1                mib2c.1                perldos.1              perlstyle5.16.1        sed.1                 zshzle.1
extcheck.1            migrateLocalKDC.1      perldos5.16.1          perlstyle5.18.1        seeksize.d.1m
eyapp.1               mkdir.1                perldos5.18.1          perlsub.1              sendmail.1
eyapp5.16.1           mkfifo.1               perldsc.1              perlsub5.16.1          seq.1
[wansongHome@wansong:/usr/share/man/man1]$ cat man
man.1      manpath.1
[wansongHome@wansong:/usr/share/man/man1]$ cat ba
base64.1    basename.1  bash.1      bashbug.1   batch.1
[wansongHome@wansong:/usr/share/man/man1]$ cat ba
base64.1    basename.1  bash.1      bashbug.1   batch.1
[wansongHome@wansong:/usr/share/man/man1]$ cat ba
base64.1    basename.1  bash.1      bashbug.1   batch.1
[wansongHome@wansong:/usr/share/man/man1]$ cat bas
base64.1    basename.1  bash.1      bashbug.1
[wansongHome@wansong:/usr/share/man/man1]$ cat bas
base64.1    basename.1  bash.1      bashbug.1
[wansongHome@wansong:/usr/share/man/man1]$ cat bas
base64.1    basename.1  bash.1      bashbug.1
[wansongHome@wansong:/usr/share/man/man1]$ cat bash
bash.1      bashbug.1
[wansongHome@wansong:/usr/share/man/man1]$ cat bash
bash.1      bashbug.1
[wansongHome@wansong:/usr/share/man/man1]$ cat bash
bash.1      bashbug.1
[wansongHome@wansong:/usr/share/man/man1]$ cd
[wansongHome@wansong:~]$ man bash > bash.man.txt
[wansongHome@wansong:~]$ open bash.man.txt
[wansongHome@wansong:~]$ rm bash.man.txt
[wansongHome@wansong:~]$ man bash
[wansongHome@wansong:~]$ vim test.sh
[wansongHome@wansong:~]$ ./test.sh
./test.sh
we are being executed
[wansongHome@wansong:~]$ man bash
[wansongHome@wansong:~]$ source  test.sh
bash
we are being sourced
[wansongHome@wansong:~]$ bash test.sh
/usr/local/bin/bash
we are being sourced
[wansongHome@wansong:~]$ vim test.sh
[wansongHome@wansong:~]$ ./test.sh
./test.sh, ./test.sh
we are being sourced
[wansongHome@wansong:~]$ bash test.sh
/usr/local/bin/bash, test.sh
we are being sourced
[wansongHome@wansong:~]$ source  test.sh
test.sh, -bash
we are being sourced
[wansongHome@wansong:~]$ man bash
```

              Individual _readline_ initialization file

AUTHORS
       Brian Fox, Free Software Foundation