```
**********************************************************
   45302 Sat Jul 19 14:19:57 2014
new/exception_lists/cstyle
Latest round of fixes per RM and AL.  Fix bugs found in man.c.
**********************************************************
```

```
  1 usr/src/cmd/krb5/kadmin/cli/kadmin_ct.c
  2 usr/src/cmd/krb5/kadmin/cli/kadmin.c
  3 usr/src/cmd/krb5/kadmin/cli/kadmin.h
  4 usr/src/cmd/krb5/kadmin/cli/keytab.c
  5 usr/src/cmd/krb5/kadmin/cli/ss_wrapper.c
  6 usr/src/cmd/krb5/kadmin/dbutil/dump.c
  7 usr/src/cmd/krb5/kadmin/dbutil/import_err.h
  8 usr/src/cmd/krb5/kadmin/dbutil/kadm5_create.c
  9 usr/src/cmd/krb5/kadmin/dbutil/kdb5_create.c
 10 usr/src/cmd/krb5/kadmin/dbutil/kdb5_destroy.c
 11 usr/src/cmd/krb5/kadmin/dbutil/kdb5_stash.c
 12 usr/src/cmd/krb5/kadmin/dbutil/kdb5_util.c
 13 usr/src/cmd/krb5/kadmin/dbutil/kdb5_util.h
 14 usr/src/cmd/krb5/kadmin/dbutil/nstrtok.h
 15 usr/src/cmd/krb5/kadmin/dbutil/ovload.c
 16 usr/src/cmd/krb5/kadmin/dbutil/string_table.c
 17 usr/src/cmd/krb5/kadmin/dbutil/string_table.h
 18 usr/src/cmd/krb5/kadmin/dbutil/strtok.c
 19 usr/src/cmd/krb5/kadmin/dbutil/util.c
 20 usr/src/cmd/krb5/kadmin/kpasswd/kpasswd_strings.h
 21 usr/src/cmd/krb5/kadmin/kpasswd/kpasswd.c
 22 usr/src/cmd/krb5/kadmin/kpasswd/kpasswd.h
 23 usr/src/cmd/krb5/kadmin/kpasswd/tty_kpasswd.c
 24 usr/src/cmd/krb5/kadmin/ktutil/ktutil_ct.c
 25 usr/src/cmd/krb5/kadmin/ktutil/ktutil_funcs.c
 26 usr/src/cmd/krb5/kadmin/ktutil/ktutil.c
 27 usr/src/cmd/krb5/kadmin/ktutil/ktutil.h
 28 usr/src/cmd/krb5/kadmin/server/kadm_rpc_svc.c
 29 usr/src/cmd/krb5/kadmin/server/misc.c
 30 usr/src/cmd/krb5/kadmin/server/misc.h
 31 usr/src/cmd/krb5/kadmin/server/ovsec_kadmd.c
 32 usr/src/cmd/krb5/kadmin/server/server_glue_v1.c
 33 usr/src/cmd/krb5/kadmin/server/server_stubs.c
 34 usr/src/cmd/krb5/kdestroy/kdestroy.c
 35 usr/src/cmd/krb5/kinit/kinit.c
 36 usr/src/cmd/krb5/klist/klist.c
 37 usr/src/cmd/krb5/krb5kdc/dispatch.c
 38 usr/src/cmd/krb5/krb5kdc/do_as_req.c
 39 usr/src/cmd/krb5/krb5kdc/do_tgs_req.c
 40 usr/src/cmd/krb5/krb5kdc/extern.c
 41 usr/src/cmd/krb5/krb5kdc/extern.h
 42 usr/src/cmd/krb5/krb5kdc/kdc_preauth.c
 43 usr/src/cmd/krb5/krb5kdc/kdc_util.c
 44 usr/src/cmd/krb5/krb5kdc/kdc_util.h
 45 usr/src/cmd/krb5/krb5kdc/main.c
 46 usr/src/cmd/krb5/krb5kdc/network.c
 47 usr/src/cmd/krb5/krb5kdc/policy.c
 48 usr/src/cmd/krb5/krb5kdc/policy.h
 49 usr/src/cmd/krb5/krb5kdc/replay.c
 50 usr/src/cmd/krb5/krb5kdc/sock2p.c
 51 usr/src/cmd/krb5/ldap_util/kdb5_ldap_list.c
 52 usr/src/cmd/krb5/ldap_util/kdb5_ldap_list.h
 53 usr/src/cmd/krb5/ldap_util/kdb5_ldap_policy.c
 54 usr/src/cmd/krb5/ldap_util/kdb5_ldap_policy.h
 55 usr/src/cmd/krb5/ldap_util/kdb5_ldap_realm.c
 56 usr/src/cmd/krb5/ldap_util/kdb5_ldap_realm.h
 57 usr/src/cmd/krb5/ldap_util/kdb5_ldap_services.c
 58 usr/src/cmd/krb5/ldap_util/kdb5_ldap_services.h
 59 usr/src/cmd/krb5/ldap_util/kdb5_ldap_util.c
 60 usr/src/cmd/krb5/ldap_util/kdb5_ldap_util.h
 61 usr/src/cmd/krb5/slave/kprop.c
```



```
 62 usr/src/cmd/krb5/slave/kprop.h
 63 usr/src/cmd/krb5/slave/kpropd.c
 64 usr/src/cmd/mandoc/arch.c
 65 usr/src/cmd/mandoc/att.c
 66 usr/src/cmd/mandoc/chars.c
 67 usr/src/cmd/mandoc/config.h
 68 usr/src/cmd/mandoc/eqn.c
 69 usr/src/cmd/mandoc/eqn_html.c
 70 usr/src/cmd/mandoc/eqn_term.c
 71 usr/src/cmd/mandoc/html.c
 72 usr/src/cmd/mandoc/html.h
 73 usr/src/cmd/mandoc/lib.c
 74 usr/src/cmd/mandoc/libman.h
 75 usr/src/cmd/mandoc/libmandoc.h
 76 usr/src/cmd/mandoc/libmdoc.h
 77 usr/src/cmd/mandoc/libroff.h
 78 usr/src/cmd/mandoc/main.c
 79 usr/src/cmd/mandoc/main.h
 80 usr/src/cmd/mandoc/man.c
 81 usr/src/cmd/mandoc/man.h
 82 usr/src/cmd/mandoc/man_hash.c
 83 usr/src/cmd/mandoc/man_html.c
 84 usr/src/cmd/mandoc/man_macro.c
 85 usr/src/cmd/mandoc/man_term.c
 86 usr/src/cmd/mandoc/man_validate.c
 87 usr/src/cmd/mandoc/mandoc.c
 88 usr/src/cmd/mandoc/mandoc.h
 89 usr/src/cmd/mandoc/mdoc.c
 90 usr/src/cmd/mandoc/mdoc.h
 91 usr/src/cmd/mandoc/mdoc_argv.c
 92 usr/src/cmd/mandoc/mdoc_hash.c
 93 usr/src/cmd/mandoc/mdoc_html.c
 94 usr/src/cmd/mandoc/mdoc_macro.c
 95 usr/src/cmd/mandoc/mdoc_man.c
 96 usr/src/cmd/mandoc/mdoc_term.c
 97 usr/src/cmd/mandoc/mdoc_validate.c
 98 usr/src/cmd/mandoc/msec.c
 99 usr/src/cmd/mandoc/out.c
100 usr/src/cmd/mandoc/out.h
101 usr/src/cmd/mandoc/preconv.c
102 usr/src/cmd/mandoc/read.c
103 usr/src/cmd/mandoc/roff.c
104 usr/src/cmd/mandoc/st.c
105 usr/src/cmd/mandoc/tbl.c
106 usr/src/cmd/mandoc/tbl_data.c
107 usr/src/cmd/mandoc/tbl_html.c
108 usr/src/cmd/mandoc/tbl_layout.c
109 usr/src/cmd/mandoc/tbl_opts.c
110 usr/src/cmd/mandoc/tbl_term.c
111 usr/src/cmd/mandoc/term.c
112 usr/src/cmd/mandoc/term.h
113 usr/src/cmd/mandoc/term_ascii.c
114 usr/src/cmd/mandoc/term_ps.c
115 usr/src/cmd/mandoc/tree.c
116 usr/src/cmd/mandoc/vol.c
117 usr/src/common/bzip2/bzlib.h
118 usr/src/common/bzip2/crctable.c
119 usr/src/common/bzip2/randtable.c
120 usr/src/common/bzip2/blocksort.c
121 usr/src/common/bzip2/compress.c
122 usr/src/common/bzip2/bzlib.c
123 usr/src/common/bzip2/decompress.c
124 usr/src/common/bzip2/bzlib_private.h
125 usr/src/common/bzip2/huffman.c
126 usr/src/common/openssl/crypto/krb5/krb5_asn.c
127 usr/src/common/openssl/crypto/krb5/krb5_asn.h
```

```
usr/src/lib/gss_mechs/mech_krb5/crypto/aes/aes_s2k.c
usr/src/lib/gss_mechs/mech_krb5/crypto/cksumtype_to_string.c
usr/src/lib/gss_mechs/mech_krb5/crypto/coll_proof_cksum.c
usr/src/lib/gss_mechs/mech_krb5/crypto/crc32/crc.c
usr/src/lib/gss_mechs/mech_krb5/crypto/des/afsstring2key.c
usr/src/lib/gss_mechs/mech_krb5/crypto/des/string2key.c
usr/src/lib/gss_mechs/mech_krb5/crypto/dk/stringtokey.c
usr/src/lib/gss_mechs/mech_krb5/crypto/enctype_compare.c
usr/src/lib/gss_mechs/mech_krb5/crypto/enctype_to_string.c
usr/src/lib/gss_mechs/mech_krb5/crypto/hash_provider/hash_md5.c
usr/src/lib/gss_mechs/mech_krb5/crypto/hash_provider/hash_sha1.c
usr/src/lib/gss_mechs/mech_krb5/crypto/keyed_checksum_types.c
usr/src/lib/gss_mechs/mech_krb5/crypto/keyed_cksum.c
usr/src/lib/gss_mechs/mech_krb5/crypto/keyhash_provider/hmac_md5.c
usr/src/lib/gss_mechs/mech_krb5/crypto/keyhash_provider/k5_md5des.c
usr/src/lib/gss_mechs/mech_krb5/crypto/keylengths.c
usr/src/lib/gss_mechs/mech_krb5/crypto/make_random_key.c
usr/src/lib/gss_mechs/mech_krb5/crypto/md4/md4.c
usr/src/lib/gss_mechs/mech_krb5/crypto/old_api_glue.c
usr/src/lib/gss_mechs/mech_krb5/crypto/old/des_stringtokey.c
usr/src/lib/gss_mechs/mech_krb5/crypto/pbkdf2.c
usr/src/lib/gss_mechs/mech_krb5/crypto/random_to_key.c
usr/src/lib/gss_mechs/mech_krb5/crypto/state.c
usr/src/lib/gss_mechs/mech_krb5/crypto/string_to_cksumtype.c
usr/src/lib/gss_mechs/mech_krb5/crypto/string_to_enctype.c
usr/src/lib/gss_mechs/mech_krb5/crypto/string_to_key.c
usr/src/lib/gss_mechs/mech_krb5/crypto/valid_cksumtype.c
usr/src/lib/gss_mechs/mech_krb5/crypto/valid_enctype.c
usr/src/lib/gss_mechs/mech_krb5/et/com_err.c
usr/src/lib/gss_mechs/mech_krb5/et/error_message.c
usr/src/lib/gss_mechs/mech_krb5/et/error_table.h
usr/src/lib/gss_mechs/mech_krb5/et/internal.h
usr/src/lib/gss_mechs/mech_krb5/et/mit-sipb-copyright.h
usr/src/lib/gss_mechs/mech_krb5/include/cache-addrinfo.h
usr/src/lib/gss_mechs/mech_krb5/include/cm.h
usr/src/lib/gss_mechs/mech_krb5/include/com_err.h
usr/src/lib/gss_mechs/mech_krb5/include/db-config.h
usr/src/lib/gss_mechs/mech_krb5/include/db.h
usr/src/lib/gss_mechs/mech_krb5/include/fake-addrinfo.h
usr/src/lib/gss_mechs/mech_krb5/include/foreachaddr.h
usr/src/lib/gss_mechs/mech_krb5/include/k5-int-pkinit.h
usr/src/lib/gss_mechs/mech_krb5/include/k5-utf8.h
usr/src/lib/gss_mechs/mech_krb5/include/kdb_kt.h
usr/src/lib/gss_mechs/mech_krb5/include/krb5_libinit.h
usr/src/lib/gss_mechs/mech_krb5/include/krb5/adm_defs.h
usr/src/lib/gss_mechs/mech_krb5/include/krb5/adm_proto.h
usr/src/lib/gss_mechs/mech_krb5/include/krb5/adm.h
usr/src/lib/gss_mechs/mech_krb5/include/krb5/copyright.h
usr/src/lib/gss_mechs/mech_krb5/include/krb5/k5-err.h
usr/src/lib/gss_mechs/mech_krb5/include/krb5/k5-plugin.h
usr/src/lib/gss_mechs/mech_krb5/include/krb5/kdb_dbc.h
usr/src/lib/gss_mechs/mech_krb5/include/krb5/kdb.h
usr/src/lib/gss_mechs/mech_krb5/include/locate_plugin.h
usr/src/lib/gss_mechs/mech_krb5/include/osconf.h
usr/src/lib/gss_mechs/mech_krb5/include/port-sockets.h
usr/src/lib/gss_mechs/mech_krb5/include/preauth_plugin.h
usr/src/lib/gss_mechs/mech_krb5/include/socket-utils.h
usr/src/lib/gss_mechs/mech_krb5/krb5/asn.1/asn1_decode.c
usr/src/lib/gss_mechs/mech_krb5/krb5/asn.1/asn1_decode.h
usr/src/lib/gss_mechs/mech_krb5/krb5/asn.1/asn1_encode.c
usr/src/lib/gss_mechs/mech_krb5/krb5/asn.1/asn1_encode.h
usr/src/lib/gss_mechs/mech_krb5/krb5/asn.1/asn1_get.c
usr/src/lib/gss_mechs/mech_krb5/krb5/asn.1/asn1_get.h
usr/src/lib/gss_mechs/mech_krb5/krb5/asn.1/asn1_k_decode.c
usr/src/lib/gss_mechs/mech_krb5/krb5/asn.1/asn1_k_decode.h
usr/src/lib/gss_mechs/mech_krb5/krb5/asn.1/asn1_k_encode.c
```



```
usr/src/lib/gss_mechs/mech_krb5/krb5/asn.1/asn1_k_encode.h
usr/src/lib/gss_mechs/mech_krb5/krb5/asn.1/asn1_make.c
usr/src/lib/gss_mechs/mech_krb5/krb5/asn.1/asn1_make.h
usr/src/lib/gss_mechs/mech_krb5/krb5/asn.1/asn1_misc.c
usr/src/lib/gss_mechs/mech_krb5/krb5/asn.1/asn1_misc.h
usr/src/lib/gss_mechs/mech_krb5/krb5/asn.1/asn1buf.h
usr/src/lib/gss_mechs/mech_krb5/krb5/asn.1/krb5_decode.c
usr/src/lib/gss_mechs/mech_krb5/krb5/asn.1/krb5_encode.c
usr/src/lib/gss_mechs/mech_krb5/krb5/asn.1/krbasn1.h
usr/src/lib/gss_mechs/mech_krb5/krb5/asn.1/ldap_key_seq.c
usr/src/lib/gss_mechs/mech_krb5/krb5/ccache/cc_file.c
usr/src/lib/gss_mechs/mech_krb5/krb5/ccache/cc_memory.c
usr/src/lib/gss_mechs/mech_krb5/krb5/ccache/cc_retr.c
usr/src/lib/gss_mechs/mech_krb5/krb5/ccache/cc-int.h
usr/src/lib/gss_mechs/mech_krb5/krb5/ccache/ccbase.c
usr/src/lib/gss_mechs/mech_krb5/krb5/ccache/cccopy.c
usr/src/lib/gss_mechs/mech_krb5/krb5/ccache/ccdefault.c
usr/src/lib/gss_mechs/mech_krb5/krb5/ccache/ccdefops.c
usr/src/lib/gss_mechs/mech_krb5/krb5/ccache/ccfns.c
usr/src/lib/gss_mechs/mech_krb5/krb5/ccache/fcc.h
usr/src/lib/gss_mechs/mech_krb5/krb5/ccache/scc.h
usr/src/lib/gss_mechs/mech_krb5/krb5/ccache/ser_cc.c
usr/src/lib/gss_mechs/mech_krb5/krb5/error_tables/adm_err.h
usr/src/lib/gss_mechs/mech_krb5/krb5/keytab/file/ktfile.h
usr/src/lib/gss_mechs/mech_krb5/krb5/keytab/kt_file.c
usr/src/lib/gss_mechs/mech_krb5/krb5/keytab/kt_srvtab.c
usr/src/lib/gss_mechs/mech_krb5/krb5/keytab/kt-int.h
usr/src/lib/gss_mechs/mech_krb5/krb5/keytab/ktadd.c
usr/src/lib/gss_mechs/mech_krb5/krb5/keytab/ktbase.c
usr/src/lib/gss_mechs/mech_krb5/krb5/keytab/ktdefault.c
usr/src/lib/gss_mechs/mech_krb5/krb5/keytab/ktfns.c
usr/src/lib/gss_mechs/mech_krb5/krb5/keytab/ktfr_entry.c
usr/src/lib/gss_mechs/mech_krb5/krb5/keytab/ktremove.c
usr/src/lib/gss_mechs/mech_krb5/krb5/keytab/read_servi.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/addr_comp.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/addr_order.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/addr_srch.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/appdefault.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/bld_pr_ext.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/bld_princ.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/chk_trans.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/cleanup.h
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/conv_princ.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/copy_addrs.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/copy_creds.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/copy_data.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/copy_tick.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/cp_key_cnt.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/decode_kdc.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/decrypt_tk.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/deltat.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/enc_helper.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/encode_kdc.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/encrypt_tk.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/free_rtree.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/fwd_tgt.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/gc_frm_kdc.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/gc_via_tkt.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/gen_seqnum.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/gen_subkey.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/get_creds.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/get_in_tkt.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/gic_keytab.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/gic_opt.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/gic_pwd.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/init_keyblock.c
```

```
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/int-proto.h
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/kdc_rep_dc.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/kerrs.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/mk_error.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/mk_priv.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/mk_rep.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/mk_req_ext.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/mk_req.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/mk_safe.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/pac.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/pr_to_salt.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/preauth.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/preauth2.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/princ_comp.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/rd_cred.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/rd_error.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/rd_priv.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/rd_rep.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/rd_req_dec.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/rd_req.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/rd_safe.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/recvauth.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/send_tgs.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/sendauth.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/set_realm.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/srv_rcache.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/str_conv.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/tgtname.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/valid_times.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/vic_opt.c
usr/src/lib/gss_mechs/mech_krb5/krb5/krb/walk_rtree.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/accessor.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/an_to_ln.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/ccdefname.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/changepw.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/def_realm.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/dnsglue.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/dnsglue.h
usr/src/lib/gss_mechs/mech_krb5/krb5/os/dnssrv.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/foreachaddr.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/free_hstrl.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/free_krbhs.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/full_ipadr.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/gen_port.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/gen_rname.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/genaddrs.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/get_krbhst.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/gmt_mktime.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/hostaddr.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/hst_realm.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/ktdefname.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/kuserok.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/localaddr.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/locate_kdc.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/lock_file.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/mk_faddr.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/net_read.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/net_write.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/os-proto.h
usr/src/lib/gss_mechs/mech_krb5/krb5/os/osconfig.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/port2ip.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/prompter.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/promptusr.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/read_msg.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/read_pwd.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/realm_dom.c
```



```
usr/src/lib/gss_mechs/mech_krb5/krb5/os/realm_iter.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/sendto_kdc.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/sn2princ.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/thread_safe.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/unlck_file.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/ustime.c
usr/src/lib/gss_mechs/mech_krb5/krb5/os/write_msg.c
usr/src/lib/gss_mechs/mech_krb5/krb5/posix/daemon.c
usr/src/lib/gss_mechs/mech_krb5/krb5/posix/setenv.c
usr/src/lib/gss_mechs/mech_krb5/krb5/rcache/rc_base.h
usr/src/lib/gss_mechs/mech_krb5/krb5/rcache/rc_conv.c
usr/src/lib/gss_mechs/mech_krb5/krb5/rcache/rc_io.h
usr/src/lib/gss_mechs/mech_krb5/krb5/rcache/rc_none.c
usr/src/lib/gss_mechs/mech_krb5/krb5/rcache/rc-int.h
usr/src/lib/gss_mechs/mech_krb5/krb5/rcache/rcdef.c
usr/src/lib/gss_mechs/mech_krb5/krb5/rcache/rcfns.c
usr/src/lib/gss_mechs/mech_krb5/krb5/rcache/ser_rc.c
usr/src/lib/gss_mechs/mech_krb5/mech/accept_sec_context.c
usr/src/lib/gss_mechs/mech_krb5/mech/acquire_cred_with_pw.c
usr/src/lib/gss_mechs/mech_krb5/mech/acquire_cred.c
usr/src/lib/gss_mechs/mech_krb5/mech/add_cred.c
usr/src/lib/gss_mechs/mech_krb5/mech/compare_name.c
usr/src/lib/gss_mechs/mech_krb5/mech/context_time.c
usr/src/lib/gss_mechs/mech_krb5/mech/copy_ccache.c
usr/src/lib/gss_mechs/mech_krb5/mech/disp_com_err_status.c
usr/src/lib/gss_mechs/mech_krb5/mech/disp_major_status.c
usr/src/lib/gss_mechs/mech_krb5/mech/disp_name.c
usr/src/lib/gss_mechs/mech_krb5/mech/disp_status.c
usr/src/lib/gss_mechs/mech_krb5/mech/export_name.c
usr/src/lib/gss_mechs/mech_krb5/mech/export_sec_context.c
usr/src/lib/gss_mechs/mech_krb5/mech/get_tkt_flags.c
usr/src/lib/gss_mechs/mech_krb5/mech/gss_libinit.h
usr/src/lib/gss_mechs/mech_krb5/mech/import_name.c
usr/src/lib/gss_mechs/mech_krb5/mech/indicate_mechs.c
usr/src/lib/gss_mechs/mech_krb5/mech/init_sec_context.c
usr/src/lib/gss_mechs/mech_krb5/mech/inq_context.c
usr/src/lib/gss_mechs/mech_krb5/mech/inq_cred.c
usr/src/lib/gss_mechs/mech_krb5/mech/inq_names.c
usr/src/lib/gss_mechs/mech_krb5/mech/krb5_gss_glue.c
usr/src/lib/gss_mechs/mech_krb5/mech/lucid_context.c
usr/src/lib/gss_mechs/mech_krb5/mech/oid_ops.c
usr/src/lib/gss_mechs/mech_krb5/mech/process_context_token.c
usr/src/lib/gss_mechs/mech_krb5/mech/rel_buffer.c
usr/src/lib/gss_mechs/mech_krb5/mech/rel_cred.c
usr/src/lib/gss_mechs/mech_krb5/mech/rel_name.c
usr/src/lib/gss_mechs/mech_krb5/mech/rel_oid_set.c
usr/src/lib/gss_mechs/mech_krb5/mech/rel_oid.c
usr/src/lib/gss_mechs/mech_krb5/mech/set_allowable_enctypes.c
usr/src/lib/gss_mechs/mech_krb5/mech/set_ccache.c
usr/src/lib/gss_mechs/mech_krb5/mech/util_buffer_set.c
usr/src/lib/gss_mechs/mech_krb5/mech/util_buffer.c
usr/src/lib/gss_mechs/mech_krb5/mech/util_cksum.c
usr/src/lib/gss_mechs/mech_krb5/mech/util_ctxsetup.c
usr/src/lib/gss_mechs/mech_krb5/mech/util_dup.c
usr/src/lib/gss_mechs/mech_krb5/mech/util_localhost.c
usr/src/lib/gss_mechs/mech_krb5/mech/utl_nohash_validate.c
usr/src/lib/gss_mechs/mech_krb5/profile/prof_err.h
usr/src/lib/gss_mechs/mech_krb5/profile/prof_get.c
usr/src/lib/gss_mechs/mech_krb5/profile/prof_set.c
usr/src/lib/gss_mechs/mech_krb5/support/errors.c
usr/src/lib/gss_mechs/mech_krb5/support/fake-addrinfo.c
usr/src/lib/gss_mechs/mech_krb5/support/init-addrinfo.c
usr/src/lib/gss_mechs/mech_krb5/support/plugins.c
usr/src/lib/gss_mechs/mech_krb5/support/supp-int.h
usr/src/lib/gss_mechs/mech_krb5/support/threads.c
usr/src/lib/gss_mechs/mech_krb5/support/utf8_conv.c
```

```
usr/src/lib/gss_mechs/mech_krb5/support/utf8.c
usr/src/lib/krb5/dyn/dyn_append.c
usr/src/lib/krb5/dyn/dyn_create.c
usr/src/lib/krb5/dyn/dyn_debug.c
usr/src/lib/krb5/dyn/dyn_delete.c
usr/src/lib/krb5/dyn/dyn_initzero.c
usr/src/lib/krb5/dyn/dyn_insert.c
usr/src/lib/krb5/dyn/dyn_paranoid.c
usr/src/lib/krb5/dyn/dyn_put.c
usr/src/lib/krb5/dyn/dyn_realloc.c
usr/src/lib/krb5/dyn/dyn_size.c
usr/src/lib/krb5/kadm5/admin_internal.h
usr/src/lib/krb5/kadm5/admin_xdr.h
usr/src/lib/krb5/kadm5/admin.h
usr/src/lib/krb5/kadm5/alt_prof.c
usr/src/lib/krb5/kadm5/chpass_util_strings.h
usr/src/lib/krb5/kadm5/chpass_util.c
usr/src/lib/krb5/kadm5/clnt/changepw.c
usr/src/lib/krb5/kadm5/clnt/client_handle.c
usr/src/lib/krb5/kadm5/clnt/client_init.c
usr/src/lib/krb5/kadm5/clnt/client_internal.h
usr/src/lib/krb5/kadm5/clnt/client_principal.c
usr/src/lib/krb5/kadm5/clnt/client_rpc.c
usr/src/lib/krb5/kadm5/clnt/clnt_chpass_util.c
usr/src/lib/krb5/kadm5/clnt/clnt_policy.c
usr/src/lib/krb5/kadm5/clnt/clnt_privs.c
usr/src/lib/krb5/kadm5/clnt/logger.c
usr/src/lib/krb5/kadm5/kadm_err.h
usr/src/lib/krb5/kadm5/kadm_rpc_xdr.c
usr/src/lib/krb5/kadm5/kadm_rpc.h
usr/src/lib/krb5/kadm5/misc_free.c
usr/src/lib/krb5/kadm5/server_internal.h
usr/src/lib/krb5/kadm5/srv/adb_xdr.c
usr/src/lib/krb5/kadm5/srv/chgpwd.c
usr/src/lib/krb5/kadm5/srv/logger.c
usr/src/lib/krb5/kadm5/srv/server_acl.c
usr/src/lib/krb5/kadm5/srv/server_acl.h
usr/src/lib/krb5/kadm5/srv/server_dict.c
usr/src/lib/krb5/kadm5/srv/server_handle.c
usr/src/lib/krb5/kadm5/srv/server_init.c
usr/src/lib/krb5/kadm5/srv/server_kdb.c
usr/src/lib/krb5/kadm5/srv/server_misc.c
usr/src/lib/krb5/kadm5/srv/svr_chpass_util.c
usr/src/lib/krb5/kadm5/srv/svr_iters.c
usr/src/lib/krb5/kadm5/srv/svr_misc_free.c
usr/src/lib/krb5/kadm5/srv/svr_policy.c
usr/src/lib/krb5/kadm5/srv/svr_principal.c
usr/src/lib/krb5/kadm5/srv/xdr_alloc.c
usr/src/lib/krb5/kadm5/str_conv.c
usr/src/lib/krb5/kdb/adb_err.h
usr/src/lib/krb5/kdb/decrypt_key.c
usr/src/lib/krb5/kdb/encrypt_key.c
usr/src/lib/krb5/kdb/kdb_cpw.c
usr/src/lib/krb5/kdb/kdb_default.c
usr/src/lib/krb5/kdb/kdb5.c
usr/src/lib/krb5/kdb/kdb5.h
usr/src/lib/krb5/kdb/keytab.c
usr/src/lib/krb5/plugins/kdb/db2/adb_openclose.c
usr/src/lib/krb5/plugins/kdb/db2/adb_policy.c
usr/src/lib/krb5/plugins/kdb/db2/db2_exp.c
usr/src/lib/krb5/plugins/kdb/db2/kdb_compat.h
usr/src/lib/krb5/plugins/kdb/db2/kdb_db2.c
usr/src/lib/krb5/plugins/kdb/db2/kdb_db2.h
usr/src/lib/krb5/plugins/kdb/db2/kdb_xdr.c
usr/src/lib/krb5/plugins/kdb/db2/kdb_xdr.h
usr/src/lib/krb5/plugins/kdb/db2/libdb2/btree/bt_close.c
```



```
usr/src/lib/krb5/plugins/kdb/db2/libdb2/btree/bt_conv.c
usr/src/lib/krb5/plugins/kdb/db2/libdb2/btree/bt_debug.c
usr/src/lib/krb5/plugins/kdb/db2/libdb2/btree/bt_delete.c
usr/src/lib/krb5/plugins/kdb/db2/libdb2/btree/bt_get.c
usr/src/lib/krb5/plugins/kdb/db2/libdb2/btree/bt_open.c
usr/src/lib/krb5/plugins/kdb/db2/libdb2/btree/bt_overflow.c
usr/src/lib/krb5/plugins/kdb/db2/libdb2/btree/bt_page.c
usr/src/lib/krb5/plugins/kdb/db2/libdb2/btree/bt_put.c
usr/src/lib/krb5/plugins/kdb/db2/libdb2/btree/bt_search.c
usr/src/lib/krb5/plugins/kdb/db2/libdb2/btree/bt_seq.c
usr/src/lib/krb5/plugins/kdb/db2/libdb2/btree/bt_split.c
usr/src/lib/krb5/plugins/kdb/db2/libdb2/btree/bt_utils.c
usr/src/lib/krb5/plugins/kdb/db2/libdb2/btree/btree.h
usr/src/lib/krb5/plugins/kdb/db2/libdb2/btree/extern.h
usr/src/lib/krb5/plugins/kdb/db2/libdb2/db/db.c
usr/src/lib/krb5/plugins/kdb/db2/libdb2/hash/dbm.c
usr/src/lib/krb5/plugins/kdb/db2/libdb2/hash/extern.h
usr/src/lib/krb5/plugins/kdb/db2/libdb2/hash/hash_bigkey.c
usr/src/lib/krb5/plugins/kdb/db2/libdb2/hash/hash_func.c
usr/src/lib/krb5/plugins/kdb/db2/libdb2/hash/hash_log2.c
usr/src/lib/krb5/plugins/kdb/db2/libdb2/hash/hash_page.c
usr/src/lib/krb5/plugins/kdb/db2/libdb2/hash/hash.c
usr/src/lib/krb5/plugins/kdb/db2/libdb2/hash/hash.h
usr/src/lib/krb5/plugins/kdb/db2/libdb2/hash/hsearch.c
usr/src/lib/krb5/plugins/kdb/db2/libdb2/hash/page.h
usr/src/lib/krb5/plugins/kdb/db2/libdb2/hash/search.h
usr/src/lib/krb5/plugins/kdb/db2/libdb2/include/db-int.h
usr/src/lib/krb5/plugins/kdb/db2/libdb2/include/db-ndbm.h
usr/src/lib/krb5/plugins/kdb/db2/libdb2/include/db-queue.h
usr/src/lib/krb5/plugins/kdb/db2/libdb2/mpool/mpool.c
usr/src/lib/krb5/plugins/kdb/db2/libdb2/mpool/mpool.h
usr/src/lib/krb5/plugins/kdb/db2/libdb2/recno/extern.h
usr/src/lib/krb5/plugins/kdb/db2/libdb2/recno/rec_close.c
usr/src/lib/krb5/plugins/kdb/db2/libdb2/recno/rec_delete.c
usr/src/lib/krb5/plugins/kdb/db2/libdb2/recno/rec_get.c
usr/src/lib/krb5/plugins/kdb/db2/libdb2/recno/rec_open.c
usr/src/lib/krb5/plugins/kdb/db2/libdb2/recno/rec_put.c
usr/src/lib/krb5/plugins/kdb/db2/libdb2/recno/rec_search.c
usr/src/lib/krb5/plugins/kdb/db2/libdb2/recno/rec_seq.c
usr/src/lib/krb5/plugins/kdb/db2/libdb2/recno/rec_utils.c
usr/src/lib/krb5/plugins/kdb/db2/libdb2/recno/recno.h
usr/src/lib/krb5/plugins/kdb/db2/pol_xdr.c
usr/src/lib/krb5/plugins/kdb/db2/policy_db.h
usr/src/lib/krb5/plugins/kdb/ldap/ldap_exp.c
usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/kdb_ldap_conn.c
usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/kdb_ldap.c
usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/kdb_ldap.h
usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/kdb_xdr.c
usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/kdb_xdr.h
usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_create.c
usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_err.c
usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_err.h
usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_fetch_mkey.c
usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_handle.c
usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_handle.h
usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_krbcontainer.c
usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_krbcontainer.h
usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_main.h
usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_misc.c
usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_misc.h
usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_principal.c
usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_principal.h
usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_principal2.c
usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_pwd_policy.c
usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_pwd_policy.h
usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_realm.c
```

```
usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_realm.h
usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_service_rights.c
usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_service_stash.c
usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_service_stash.h
usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_services.c
usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_services.h
usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_tkt_policy.c
usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_tkt_policy.h
usr/src/lib/krb5/plugins/preauth/pkinit/pkinit_accessor.c
usr/src/lib/krb5/plugins/preauth/pkinit/pkinit_accessor.h
usr/src/lib/krb5/plugins/preauth/pkinit/pkinit_clnt.c
usr/src/lib/krb5/plugins/preauth/pkinit/pkinit_crypto_openssl.c
usr/src/lib/krb5/plugins/preauth/pkinit/pkinit_crypto_openssl.h
usr/src/lib/krb5/plugins/preauth/pkinit/pkinit_crypto.h
usr/src/lib/krb5/plugins/preauth/pkinit/pkinit_identity.c
usr/src/lib/krb5/plugins/preauth/pkinit/pkinit_lib.c
usr/src/lib/krb5/plugins/preauth/pkinit/pkinit_matching.c
usr/src/lib/krb5/plugins/preauth/pkinit/pkinit_profile.c
usr/src/lib/krb5/plugins/preauth/pkinit/pkinit_srv.c
usr/src/lib/krb5/plugins/preauth/pkinit/pkinit.h
usr/src/lib/krb5/ss/copyright.h
usr/src/lib/krb5/ss/data.c
usr/src/lib/krb5/ss/error.c
usr/src/lib/krb5/ss/execute_cmd.c
usr/src/lib/krb5/ss/help.c
usr/src/lib/krb5/ss/invocation.c
usr/src/lib/krb5/ss/list_rqs.c
usr/src/lib/krb5/ss/listen.c
usr/src/lib/krb5/ss/mit-sipb-copyright.h
usr/src/lib/krb5/ss/mk_cmds.c
usr/src/lib/krb5/ss/options.c
usr/src/lib/krb5/ss/pager.c
usr/src/lib/krb5/ss/parse.c
usr/src/lib/krb5/ss/prompt.c
usr/src/lib/krb5/ss/request_tbl.c
usr/src/lib/krb5/ss/requests.c
usr/src/lib/krb5/ss/ss_internal.h
usr/src/lib/krb5/ss/ss.h
usr/src/lib/krb5/ss/std_rqs.c
usr/src/lib/krb5/ss/utils.c
usr/src/lib/libgss/g_glue.c
usr/src/lib/librstp/common/base.h
usr/src/lib/librstp/common/choose.h
usr/src/lib/librstp/common/edge.c
usr/src/lib/librstp/common/edge.h
usr/src/lib/librstp/common/migrate.c
usr/src/lib/librstp/common/migrate.h
usr/src/lib/librstp/common/p2p.c
usr/src/lib/librstp/common/p2p.h
usr/src/lib/librstp/common/pcost.c
usr/src/lib/librstp/common/pcost.h
usr/src/lib/librstp/common/port.c
usr/src/lib/librstp/common/port.h
usr/src/lib/librstp/common/portinfo.c
usr/src/lib/librstp/common/portinfo.h
usr/src/lib/librstp/common/rolesel.c
usr/src/lib/librstp/common/rolesel.h
usr/src/lib/librstp/common/roletrns.c
usr/src/lib/librstp/common/roletrns.h
usr/src/lib/librstp/common/statmch.c
usr/src/lib/librstp/common/statmch.h
usr/src/lib/librstp/common/stp_bpdu.h
usr/src/lib/librstp/common/stp_in.c
usr/src/lib/librstp/common/stp_in.h
usr/src/lib/librstp/common/stp_to.h
usr/src/lib/librstp/common/stp_vectors.h
```



```
usr/src/lib/librstp/common/stpm.c
usr/src/lib/librstp/common/stpm.h
usr/src/lib/librstp/common/stpmgmt.c
usr/src/lib/librstp/common/sttrans.c
usr/src/lib/librstp/common/sttrans.h
usr/src/lib/librstp/common/times.c
usr/src/lib/librstp/common/times.h
usr/src/lib/librstp/common/topoch.c
usr/src/lib/librstp/common/topoch.h
usr/src/lib/librstp/common/transmit.c
usr/src/lib/librstp/common/transmit.h
usr/src/lib/librstp/common/uid_stp.h
usr/src/lib/librstp/common/vector.c
usr/src/lib/librstp/common/vector.h
usr/src/uts/common/gssapi/gssapi.h
usr/src/uts/common/gssapi/mechs/krb5/crypto/block_size.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/checksum_length.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/cksumtypes.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/combine_keys.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/crc32/crc32.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/decrypt.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/default_state.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/des/d3_cbc.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/des/f_cbc.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/des/f_parity.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/des/weak_key.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/dk/checksum.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/dk/derive.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/dk/dk_decrypt.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/dk/dk_encrypt.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/enc_provider/arcfour_provider.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/enc_provider/des.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/enc_provider/des3.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/encrypt_length.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/encrypt.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/etypes.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/hash_provider/hash_crc32.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/hash_provider/hash_kmd5.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/hash_provider/hash_ksha1.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/hmac.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/keyhash_provider/descbc.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/keyhash_provider/k_hmac_md5.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/keyhash_provider/k5_kmd5des.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/make_checksum.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/mandatory_sumtype.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/nfold.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/old/old_decrypt.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/old/old_encrypt.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/prng.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/raw/raw_decrypt.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/raw/raw_encrypt.c
usr/src/uts/common/gssapi/mechs/krb5/crypto/verify_checksum.c
usr/src/uts/common/gssapi/mechs/krb5/include/aes_s2k.h
usr/src/uts/common/gssapi/mechs/krb5/include/auth_con.h
usr/src/uts/common/gssapi/mechs/krb5/include/cksumtypes.h
usr/src/uts/common/gssapi/mechs/krb5/include/crc-32.h
usr/src/uts/common/gssapi/mechs/krb5/include/des_int.h
usr/src/uts/common/gssapi/mechs/krb5/include/dk.h
usr/src/uts/common/gssapi/mechs/krb5/include/enc_provider.h
usr/src/uts/common/gssapi/mechs/krb5/include/etypes.h
usr/src/uts/common/gssapi/mechs/krb5/include/gssapi_generic.h
usr/src/uts/common/gssapi/mechs/krb5/include/gssapi_krb5.h
usr/src/uts/common/gssapi/mechs/krb5/include/gssapiP_generic.h
usr/src/uts/common/gssapi/mechs/krb5/include/gssapiP_krb5.h
usr/src/uts/common/gssapi/mechs/krb5/include/hash_provider.h
usr/src/uts/common/gssapi/mechs/krb5/include/k5-int.h
```

```
usr/src/uts/common/gssapi/mechs/krb5/include/k5-platform-load_16.h
usr/src/uts/common/gssapi/mechs/krb5/include/k5-platform-load_32.h
usr/src/uts/common/gssapi/mechs/krb5/include/k5-platform-load_64.h
usr/src/uts/common/gssapi/mechs/krb5/include/k5-platform-store_16.h
usr/src/uts/common/gssapi/mechs/krb5/include/k5-platform-store_32.h
usr/src/uts/common/gssapi/mechs/krb5/include/k5-platform-store_64.h
usr/src/uts/common/gssapi/mechs/krb5/include/k5-platform.h
usr/src/uts/common/gssapi/mechs/krb5/include/k5-thread.h
usr/src/uts/common/gssapi/mechs/krb5/include/keyhash_provider.h
usr/src/uts/common/gssapi/mechs/krb5/include/krb5.h
usr/src/uts/common/gssapi/mechs/krb5/include/old.h
usr/src/uts/common/gssapi/mechs/krb5/include/raw.h
usr/src/uts/common/gssapi/mechs/krb5/include/rsa-md4.h
usr/src/uts/common/gssapi/mechs/krb5/krb5/krb/copy_athctr.c
usr/src/uts/common/gssapi/mechs/krb5/krb5/krb/copy_auth.c
usr/src/uts/common/gssapi/mechs/krb5/krb5/krb/copy_cksum.c
usr/src/uts/common/gssapi/mechs/krb5/krb5/krb/copy_key.c
usr/src/uts/common/gssapi/mechs/krb5/krb5/krb/copy_princ.c
usr/src/uts/common/gssapi/mechs/krb5/krb5/krb/init_ctx.c
usr/src/uts/common/gssapi/mechs/krb5/krb5/krb/kfree.c
usr/src/uts/common/gssapi/mechs/krb5/krb5/krb/parse.c
usr/src/uts/common/gssapi/mechs/krb5/krb5/krb/ser_actx.c
usr/src/uts/common/gssapi/mechs/krb5/krb5/krb/ser_adata.c
usr/src/uts/common/gssapi/mechs/krb5/krb5/krb/ser_addr.c
usr/src/uts/common/gssapi/mechs/krb5/krb5/krb/ser_auth.c
usr/src/uts/common/gssapi/mechs/krb5/krb5/krb/ser_cksum.c
usr/src/uts/common/gssapi/mechs/krb5/krb5/krb/ser_ctx.c
usr/src/uts/common/gssapi/mechs/krb5/krb5/krb/ser_key.c
usr/src/uts/common/gssapi/mechs/krb5/krb5/krb/ser_princ.c
usr/src/uts/common/gssapi/mechs/krb5/krb5/krb/serialize.c
usr/src/uts/common/gssapi/mechs/krb5/krb5/krb/unparse.c
usr/src/uts/common/gssapi/mechs/krb5/krb5/os/c_ustime.c
usr/src/uts/common/gssapi/mechs/krb5/krb5/os/init_os_ctx.c
usr/src/uts/common/gssapi/mechs/krb5/krb5/os/timeofday.c
usr/src/uts/common/gssapi/mechs/krb5/krb5/os/toffset.c
usr/src/uts/common/gssapi/mechs/krb5/mech/delete_sec_context.c
usr/src/uts/common/gssapi/mechs/krb5/mech/gssapi_krb5.c
usr/src/uts/common/gssapi/mechs/krb5/mech/import_sec_context.c
usr/src/uts/common/gssapi/mechs/krb5/mech/k5seal.c
usr/src/uts/common/gssapi/mechs/krb5/mech/k5sealv3.c
usr/src/uts/common/gssapi/mechs/krb5/mech/k5unseal.c
usr/src/uts/common/gssapi/mechs/krb5/mech/seal.c
usr/src/uts/common/gssapi/mechs/krb5/mech/ser_sctx.c
usr/src/uts/common/gssapi/mechs/krb5/mech/sign.c
usr/src/uts/common/gssapi/mechs/krb5/mech/unseal.c
usr/src/uts/common/gssapi/mechs/krb5/mech/util_crypt.c
usr/src/uts/common/gssapi/mechs/krb5/mech/util_ordering.c
usr/src/uts/common/gssapi/mechs/krb5/mech/util_seed.c
usr/src/uts/common/gssapi/mechs/krb5/mech/util_seqnum.c
usr/src/uts/common/gssapi/mechs/krb5/mech/util_set.c
usr/src/uts/common/gssapi/mechs/krb5/mech/util_token.c
usr/src/uts/common/gssapi/mechs/krb5/mech/util_validate.c
usr/src/uts/common/gssapi/mechs/krb5/mech/val_cred.c
usr/src/uts/common/gssapi/mechs/krb5/mech/verify.c
usr/src/uts/common/gssapi/mechs/krb5/mech/wrap_size_limit.c
usr/src/uts/common/io/e1000api/e1000_80003es2lan.c
usr/src/uts/common/io/e1000api/e1000_80003es2lan.h
usr/src/uts/common/io/e1000api/e1000_82540.c
usr/src/uts/common/io/e1000api/e1000_82541.c
usr/src/uts/common/io/e1000api/e1000_82541.h
usr/src/uts/common/io/e1000api/e1000_82542.c
usr/src/uts/common/io/e1000api/e1000_82543.c
usr/src/uts/common/io/e1000api/e1000_82543.h
usr/src/uts/common/io/e1000api/e1000_82571.c
usr/src/uts/common/io/e1000api/e1000_82571.h
usr/src/uts/common/io/e1000api/e1000_82575.c
```



```
usr/src/uts/common/io/e1000api/e1000_82575.h
usr/src/uts/common/io/e1000api/e1000_api.c
usr/src/uts/common/io/e1000api/e1000_api.h
usr/src/uts/common/io/e1000api/e1000_defines.h
usr/src/uts/common/io/e1000api/e1000_hw.h
usr/src/uts/common/io/e1000api/e1000_i210.c
usr/src/uts/common/io/e1000api/e1000_i210.h
usr/src/uts/common/io/e1000api/e1000_ich8lan.c
usr/src/uts/common/io/e1000api/e1000_ich8lan.h
usr/src/uts/common/io/e1000api/e1000_mac.c
usr/src/uts/common/io/e1000api/e1000_mac.h
usr/src/uts/common/io/e1000api/e1000_manage.c
usr/src/uts/common/io/e1000api/e1000_manage.h
usr/src/uts/common/io/e1000api/e1000_mbx.c
usr/src/uts/common/io/e1000api/e1000_mbx.h
usr/src/uts/common/io/e1000api/e1000_nvm.c
usr/src/uts/common/io/e1000api/e1000_nvm.h
usr/src/uts/common/io/e1000api/e1000_phy.c
usr/src/uts/common/io/e1000api/e1000_phy.h
usr/src/uts/common/io/e1000api/e1000_regs.h
usr/src/uts/common/io/e1000api/e1000_vf.c
usr/src/uts/common/io/e1000api/e1000_vf.h
usr/src/uts/common/io/ixgbe/ixgbe_82598.c
usr/src/uts/common/io/ixgbe/ixgbe_82598.h
usr/src/uts/common/io/ixgbe/ixgbe_82599.c
usr/src/uts/common/io/ixgbe/ixgbe_82599.h
usr/src/uts/common/io/ixgbe/ixgbe_api.c
usr/src/uts/common/io/ixgbe/ixgbe_api.h
usr/src/uts/common/io/ixgbe/ixgbe_common.c
usr/src/uts/common/io/ixgbe/ixgbe_common.h
usr/src/uts/common/io/ixgbe/ixgbe_mbx.c
usr/src/uts/common/io/ixgbe/ixgbe_mbx.h
usr/src/uts/common/io/ixgbe/ixgbe_osdep.h
usr/src/uts/common/io/ixgbe/ixgbe_phy.c
usr/src/uts/common/io/ixgbe/ixgbe_phy.h
usr/src/uts/common/io/ixgbe/ixgbe_type.h
usr/src/uts/common/io/ixgbe/ixgbe_x540.c
usr/src/uts/common/io/ixgbe/ixgbe_x540.h
usr/src/uts/intel/io/acpica/debugger/dbcmds.c
usr/src/uts/intel/io/acpica/debugger/dbdisply.c
usr/src/uts/intel/io/acpica/debugger/dbexec.c
usr/src/uts/intel/io/acpica/debugger/dbfileio.c
usr/src/uts/intel/io/acpica/debugger/dbhistry.c
usr/src/uts/intel/io/acpica/debugger/dbinput.c
usr/src/uts/intel/io/acpica/debugger/dbmethod.c
usr/src/uts/intel/io/acpica/debugger/dbnames.c
usr/src/uts/intel/io/acpica/debugger/dbstats.c
usr/src/uts/intel/io/acpica/debugger/dbutils.c
usr/src/uts/intel/io/acpica/debugger/dbxface.c
usr/src/uts/intel/io/acpica/disassembler/dmbuffer.c
usr/src/uts/intel/io/acpica/disassembler/dmnames.c
usr/src/uts/intel/io/acpica/disassembler/dmobject.c
usr/src/uts/intel/io/acpica/disassembler/dmopcode.c
usr/src/uts/intel/io/acpica/disassembler/dmresrc.c
usr/src/uts/intel/io/acpica/disassembler/dmresrcl.c
usr/src/uts/intel/io/acpica/disassembler/dmresrcs.c
usr/src/uts/intel/io/acpica/disassembler/dmutils.c
usr/src/uts/intel/io/acpica/disassembler/dmwalk.c
usr/src/uts/intel/io/acpica/dispatcher/dsargs.c
usr/src/uts/intel/io/acpica/dispatcher/dscontrol.c
usr/src/uts/intel/io/acpica/dispatcher/dsfield.c
usr/src/uts/intel/io/acpica/dispatcher/dsinit.c
usr/src/uts/intel/io/acpica/dispatcher/dsmethod.c
usr/src/uts/intel/io/acpica/dispatcher/dsmthdat.c
usr/src/uts/intel/io/acpica/dispatcher/dsobject.c
usr/src/uts/intel/io/acpica/dispatcher/dsopcode.c
```

```
usr/src/uts/intel/io/acpica/dispatcher/dsutils.c
usr/src/uts/intel/io/acpica/dispatcher/dswexec.c
usr/src/uts/intel/io/acpica/dispatcher/dswload.c
usr/src/uts/intel/io/acpica/dispatcher/dswload2.c
usr/src/uts/intel/io/acpica/dispatcher/dswscope.c
usr/src/uts/intel/io/acpica/dispatcher/dswstate.c
usr/src/uts/intel/io/acpica/events/evevent.c
usr/src/uts/intel/io/acpica/events/evglock.c
usr/src/uts/intel/io/acpica/events/evgpe.c
usr/src/uts/intel/io/acpica/events/evgpeblk.c
usr/src/uts/intel/io/acpica/events/evgpeinit.c
usr/src/uts/intel/io/acpica/events/evgpeutil.c
usr/src/uts/intel/io/acpica/events/evmisc.c
usr/src/uts/intel/io/acpica/events/evregion.c
usr/src/uts/intel/io/acpica/events/evrgnini.c
usr/src/uts/intel/io/acpica/events/evsci.c
usr/src/uts/intel/io/acpica/events/evxface.c
usr/src/uts/intel/io/acpica/events/evxfevnt.c
usr/src/uts/intel/io/acpica/events/evxfgpe.c
usr/src/uts/intel/io/acpica/events/evxfregn.c
usr/src/uts/intel/io/acpica/executer/exconfig.c
usr/src/uts/intel/io/acpica/executer/exconvrt.c
usr/src/uts/intel/io/acpica/executer/excreate.c
usr/src/uts/intel/io/acpica/executer/exdebug.c
usr/src/uts/intel/io/acpica/executer/exdump.c
usr/src/uts/intel/io/acpica/executer/exfield.c
usr/src/uts/intel/io/acpica/executer/exfldio.c
usr/src/uts/intel/io/acpica/executer/exmisc.c
usr/src/uts/intel/io/acpica/executer/exmutex.c
usr/src/uts/intel/io/acpica/executer/exnames.c
usr/src/uts/intel/io/acpica/executer/exoparg1.c
usr/src/uts/intel/io/acpica/executer/exoparg2.c
usr/src/uts/intel/io/acpica/executer/exoparg3.c
usr/src/uts/intel/io/acpica/executer/exoparg6.c
usr/src/uts/intel/io/acpica/executer/exprep.c
usr/src/uts/intel/io/acpica/executer/exregion.c
usr/src/uts/intel/io/acpica/executer/exresnte.c
usr/src/uts/intel/io/acpica/executer/exresolv.c
usr/src/uts/intel/io/acpica/executer/exresop.c
usr/src/uts/intel/io/acpica/executer/exstore.c
usr/src/uts/intel/io/acpica/executer/exstoren.c
usr/src/uts/intel/io/acpica/executer/exstorob.c
usr/src/uts/intel/io/acpica/executer/exsystem.c
usr/src/uts/intel/io/acpica/executer/exutils.c
usr/src/uts/intel/io/acpica/hardware/hwacpi.c
usr/src/uts/intel/io/acpica/hardware/hwgpe.c
usr/src/uts/intel/io/acpica/hardware/hwpci.c
usr/src/uts/intel/io/acpica/hardware/hwregs.c
usr/src/uts/intel/io/acpica/hardware/hwsleep.c
usr/src/uts/intel/io/acpica/hardware/hwtimer.c
usr/src/uts/intel/io/acpica/hardware/hwvalid.c
usr/src/uts/intel/io/acpica/hardware/hwxface.c
usr/src/uts/intel/io/acpica/namespace/nsaccess.c
usr/src/uts/intel/io/acpica/namespace/nsalloc.c
usr/src/uts/intel/io/acpica/namespace/nsdump.c
usr/src/uts/intel/io/acpica/namespace/nsdumpdv.c
usr/src/uts/intel/io/acpica/namespace/nseval.c
usr/src/uts/intel/io/acpica/namespace/nsinit.c
usr/src/uts/intel/io/acpica/namespace/nsload.c
usr/src/uts/intel/io/acpica/namespace/nsnames.c
usr/src/uts/intel/io/acpica/namespace/nsobject.c
usr/src/uts/intel/io/acpica/namespace/nsparse.c
usr/src/uts/intel/io/acpica/namespace/nspredef.c
usr/src/uts/intel/io/acpica/namespace/nsrepair.c
usr/src/uts/intel/io/acpica/namespace/nsrepair2.c
usr/src/uts/intel/io/acpica/namespace/nssearch.c
```



```
usr/src/uts/intel/io/acpica/namespace/nsutils.c
usr/src/uts/intel/io/acpica/namespace/nswalk.c
usr/src/uts/intel/io/acpica/namespace/nsxfeval.c
usr/src/uts/intel/io/acpica/namespace/nsxfname.c
usr/src/uts/intel/io/acpica/namespace/nsxfobj.c
usr/src/uts/intel/io/acpica/parser/psargs.c
usr/src/uts/intel/io/acpica/parser/psloop.c
usr/src/uts/intel/io/acpica/parser/psopcode.c
usr/src/uts/intel/io/acpica/parser/psparse.c
usr/src/uts/intel/io/acpica/parser/psscope.c
usr/src/uts/intel/io/acpica/parser/pstree.c
usr/src/uts/intel/io/acpica/parser/psutils.c
usr/src/uts/intel/io/acpica/parser/pswalk.c
usr/src/uts/intel/io/acpica/parser/psxface.c
usr/src/uts/intel/io/acpica/resources/rsaddr.c
usr/src/uts/intel/io/acpica/resources/rscalc.c
usr/src/uts/intel/io/acpica/resources/rscreate.c
usr/src/uts/intel/io/acpica/resources/rsdump.c
usr/src/uts/intel/io/acpica/resources/rsinfo.c
usr/src/uts/intel/io/acpica/resources/rsio.c
usr/src/uts/intel/io/acpica/resources/rsirq.c
usr/src/uts/intel/io/acpica/resources/rslist.c
usr/src/uts/intel/io/acpica/resources/rsmemory.c
usr/src/uts/intel/io/acpica/resources/rsmisc.c
usr/src/uts/intel/io/acpica/resources/rsutils.c
usr/src/uts/intel/io/acpica/resources/rsxface.c
usr/src/uts/intel/io/acpica/tables/tbfadt.c
usr/src/uts/intel/io/acpica/tables/tbfind.c
usr/src/uts/intel/io/acpica/tables/tbinstal.c
usr/src/uts/intel/io/acpica/tables/tbutils.c
usr/src/uts/intel/io/acpica/tables/tbxface.c
usr/src/uts/intel/io/acpica/tables/tbxfroot.c
usr/src/uts/intel/io/acpica/utilities/utalloc.c
usr/src/uts/intel/io/acpica/utilities/utcache.c
usr/src/uts/intel/io/acpica/utilities/utclib.c
usr/src/uts/intel/io/acpica/utilities/utcopy.c
usr/src/uts/intel/io/acpica/utilities/utdebug.c
usr/src/uts/intel/io/acpica/utilities/utdecode.c
usr/src/uts/intel/io/acpica/utilities/utdelete.c
usr/src/uts/intel/io/acpica/utilities/uteval.c
usr/src/uts/intel/io/acpica/utilities/utglobal.c
usr/src/uts/intel/io/acpica/utilities/utids.c
usr/src/uts/intel/io/acpica/utilities/utinit.c
usr/src/uts/intel/io/acpica/utilities/utlock.c
usr/src/uts/intel/io/acpica/utilities/utmath.c
usr/src/uts/intel/io/acpica/utilities/utmisc.c
usr/src/uts/intel/io/acpica/utilities/utmutex.c
usr/src/uts/intel/io/acpica/utilities/utobject.c
usr/src/uts/intel/io/acpica/utilities/utosi.c
usr/src/uts/intel/io/acpica/utilities/utresrc.c
usr/src/uts/intel/io/acpica/utilities/utstate.c
usr/src/uts/intel/io/acpica/utilities/uttrack.c
usr/src/uts/intel/io/acpica/utilities/utxface.c
usr/src/uts/intel/io/acpica/utilities/utxferror.c
usr/src/uts/intel/sys/acpi/acapps.h
usr/src/uts/intel/sys/acpi/accommon.h
usr/src/uts/intel/sys/acpi/acconfig.h
usr/src/uts/intel/sys/acpi/acdebug.h
usr/src/uts/intel/sys/acpi/acdisasm.h
usr/src/uts/intel/sys/acpi/acdispat.h
usr/src/uts/intel/sys/acpi/acevents.h
usr/src/uts/intel/sys/acpi/acexcep.h
usr/src/uts/intel/sys/acpi/acglobal.h
usr/src/uts/intel/sys/acpi/achware.h
usr/src/uts/intel/sys/acpi/acinterp.h
usr/src/uts/intel/sys/acpi/aclocal.h
```

```
 920 usr/src/uts/intel/sys/acpi/acmacros.h
 921 usr/src/uts/intel/sys/acpi/acnames.h
 922 usr/src/uts/intel/sys/acpi/acnamesp.h
 923 usr/src/uts/intel/sys/acpi/acobject.h
 924 usr/src/uts/intel/sys/acpi/acopcode.h
 925 usr/src/uts/intel/sys/acpi/acoutput.h
 926 usr/src/uts/intel/sys/acpi/acparser.h
 927 usr/src/uts/intel/sys/acpi/acpi.h
 928 usr/src/uts/intel/sys/acpi/acpiosxf.h
 929 usr/src/uts/intel/sys/acpi/acpixf.h
 930 usr/src/uts/intel/sys/acpi/acpredef.h
 931 usr/src/uts/intel/sys/acpi/acresrc.h
 932 usr/src/uts/intel/sys/acpi/acrestyp.h
 933 usr/src/uts/intel/sys/acpi/acstruct.h
 934 usr/src/uts/intel/sys/acpi/actables.h
 935 usr/src/uts/intel/sys/acpi/actbl.h
 936 usr/src/uts/intel/sys/acpi/actbl1.h
 937 usr/src/uts/intel/sys/acpi/actbl2.h
 938 usr/src/uts/intel/sys/acpi/actypes.h
 939 usr/src/uts/intel/sys/acpi/acutils.h
 940 usr/src/uts/intel/sys/acpi/amlcode.h
 941 usr/src/uts/intel/sys/acpi/amlresrc.h
 942 usr/src/uts/intel/sys/acpi/platform/accygwin.h
 943 usr/src/uts/intel/sys/acpi/platform/acefi.h
 944 usr/src/uts/intel/sys/acpi/platform/acenv.h
 945 usr/src/uts/intel/sys/acpi/platform/acfreebsd.h
 946 usr/src/uts/intel/sys/acpi/platform/acgcc.h
 947 usr/src/uts/intel/sys/acpi/platform/acintel.h
 948 usr/src/uts/intel/sys/acpi/platform/aclinux.h
 949 usr/src/uts/intel/sys/acpi/platform/acmsvc.h
 950 usr/src/uts/intel/sys/acpi/platform/acnetbsd.h
 951 usr/src/uts/intel/sys/acpi/platform/acos2.h
 952 usr/src/uts/intel/sys/acpi/platform/acsolaris.h
 953 usr/src/uts/intel/sys/acpi/platform/acwin.h
 954 usr/src/uts/intel/sys/acpi/platform/acwin64.h
```

```
*******************************************************
    9208 Sat Jul 19 14:19:57 2014
new/exception_lists/hdrchk
Latest round of fixes per RM and AL.  Fix bugs found in man.c.
*******************************************************
```

```
  1 usr/src/cmd/krb5/kadmin/cli/kadmin.h
  2 usr/src/cmd/krb5/kadmin/dbutil/import_err.h
  3 usr/src/cmd/krb5/kadmin/dbutil/kdb5_util.h
  4 usr/src/cmd/krb5/kadmin/dbutil/nstrtok.h
  5 usr/src/cmd/krb5/kadmin/dbutil/string_table.h
  6 usr/src/cmd/krb5/kadmin/kpasswd/kpasswd_strings.h
  7 usr/src/cmd/krb5/kadmin/kpasswd/kpasswd.h
  8 usr/src/cmd/krb5/kadmin/ktutil/ktutil.h
  9 usr/src/cmd/krb5/kadmin/server/misc.h
 10 usr/src/cmd/krb5/krb5kdc/extern.h
 11 usr/src/cmd/krb5/krb5kdc/kdc_util.h
 12 usr/src/cmd/krb5/krb5kdc/policy.h
 13 usr/src/cmd/krb5/ldap_util/kdb5_ldap_list.h
 14 usr/src/cmd/krb5/ldap_util/kdb5_ldap_policy.h
 15 usr/src/cmd/krb5/ldap_util/kdb5_ldap_realm.h
 16 usr/src/cmd/krb5/ldap_util/kdb5_ldap_services.h
 17 usr/src/cmd/krb5/ldap_util/kdb5_ldap_util.h
 18 usr/src/cmd/krb5/slave/kprop.h
 19 usr/src/cmd/localedef/localedef.h
 20 usr/src/cmd/mandoc/config.h
 21 usr/src/cmd/mandoc/html.h
 22 usr/src/cmd/mandoc/libman.h
 23 usr/src/cmd/mandoc/libmandoc.h
 24 usr/src/cmd/mandoc/libmdoc.h
 25 usr/src/cmd/mandoc/libroff.h
 26 usr/src/cmd/mandoc/main.h
 27 usr/src/cmd/mandoc/man.h
 28 usr/src/cmd/mandoc/mandoc.h
 29 usr/src/cmd/mandoc/mdoc.h
 30 usr/src/cmd/mandoc/out.h
 31 usr/src/cmd/mandoc/term.h
 32 usr/src/common/openssl/crypto/krb5/krb5_asn.h
 33 usr/src/lib/gss_mechs/mech_krb5/et/error_table.h
 34 usr/src/lib/gss_mechs/mech_krb5/et/internal.h
 35 usr/src/lib/gss_mechs/mech_krb5/et/mit-sipb-copyright.h
 36 usr/src/lib/gss_mechs/mech_krb5/include/cache-addrinfo.h
 37 usr/src/lib/gss_mechs/mech_krb5/include/cm.h
 38 usr/src/lib/gss_mechs/mech_krb5/include/com_err.h
 39 usr/src/lib/gss_mechs/mech_krb5/include/db-config.h
 40 usr/src/lib/gss_mechs/mech_krb5/include/db.h
 41 usr/src/lib/gss_mechs/mech_krb5/include/fake-addrinfo.h
 42 usr/src/lib/gss_mechs/mech_krb5/include/foreachaddr.h
 43 usr/src/lib/gss_mechs/mech_krb5/include/k5-int-pkinit.h
 44 usr/src/lib/gss_mechs/mech_krb5/include/k5-utf8.h
 45 usr/src/lib/gss_mechs/mech_krb5/include/kdb_kt.h
 46 usr/src/lib/gss_mechs/mech_krb5/include/krb5_libinit.h
 47 usr/src/lib/gss_mechs/mech_krb5/include/krb5/adm_defs.h
 48 usr/src/lib/gss_mechs/mech_krb5/include/krb5/adm_proto.h
 49 usr/src/lib/gss_mechs/mech_krb5/include/krb5/adm.h
 50 usr/src/lib/gss_mechs/mech_krb5/include/krb5/copyright.h
 51 usr/src/lib/gss_mechs/mech_krb5/include/krb5/k5-err.h
 52 usr/src/lib/gss_mechs/mech_krb5/include/krb5/k5-plugin.h
 53 usr/src/lib/gss_mechs/mech_krb5/include/krb5/kdb_dbc.h
 54 usr/src/lib/gss_mechs/mech_krb5/include/krb5/kdb.h
 55 usr/src/lib/gss_mechs/mech_krb5/include/locate_plugin.h
 56 usr/src/lib/gss_mechs/mech_krb5/include/osconf.h
 57 usr/src/lib/gss_mechs/mech_krb5/include/port-sockets.h
 58 usr/src/lib/gss_mechs/mech_krb5/include/preauth_plugin.h
 59 usr/src/lib/gss_mechs/mech_krb5/include/socket-utils.h
 60 usr/src/lib/gss_mechs/mech_krb5/krb5/asn.1/asn1_decode.h
 61 usr/src/lib/gss_mechs/mech_krb5/krb5/asn.1/asn1_encode.h
```



```
 62 usr/src/lib/gss_mechs/mech_krb5/krb5/asn.1/asn1_get.h
 63 usr/src/lib/gss_mechs/mech_krb5/krb5/asn.1/asn1_k_decode.h
 64 usr/src/lib/gss_mechs/mech_krb5/krb5/asn.1/asn1_k_encode.h
 65 usr/src/lib/gss_mechs/mech_krb5/krb5/asn.1/asn1_make.h
 66 usr/src/lib/gss_mechs/mech_krb5/krb5/asn.1/asn1_misc.h
 67 usr/src/lib/gss_mechs/mech_krb5/krb5/asn.1/asn1buf.h
 68 usr/src/lib/gss_mechs/mech_krb5/krb5/asn.1/krbasn1.h
 69 usr/src/lib/gss_mechs/mech_krb5/krb5/ccache/cc-int.h
 70 usr/src/lib/gss_mechs/mech_krb5/krb5/ccache/fcc.h
 71 usr/src/lib/gss_mechs/mech_krb5/krb5/ccache/scc.h
 72 usr/src/lib/gss_mechs/mech_krb5/krb5/error_tables/adm_err.h
 73 usr/src/lib/gss_mechs/mech_krb5/krb5/keytab/file/ktfile.h
 74 usr/src/lib/gss_mechs/mech_krb5/krb5/keytab/kt-int.h
 75 usr/src/lib/gss_mechs/mech_krb5/krb5/krb/cleanup.h
 76 usr/src/lib/gss_mechs/mech_krb5/krb5/krb/int-proto.h
 77 usr/src/lib/gss_mechs/mech_krb5/krb5/os/dnsglue.h
 78 usr/src/lib/gss_mechs/mech_krb5/krb5/os/os-proto.h
 79 usr/src/lib/gss_mechs/mech_krb5/krb5/rcache/rc_base.h
 80 usr/src/lib/gss_mechs/mech_krb5/krb5/rcache/rc_io.h
 81 usr/src/lib/gss_mechs/mech_krb5/krb5/rcache/rc-int.h
 82 usr/src/lib/gss_mechs/mech_krb5/mech/gss_libinit.h
 83 usr/src/lib/gss_mechs/mech_krb5/profile/prof_err.h
 84 usr/src/lib/gss_mechs/mech_krb5/support/supp-int.h
 85 usr/src/lib/krb5/kadm5/admin_internal.h
 86 usr/src/lib/krb5/kadm5/admin_xdr.h
 87 usr/src/lib/krb5/kadm5/admin.h
 88 usr/src/lib/krb5/kadm5/chpass_util_strings.h
 89 usr/src/lib/krb5/kadm5/clnt/client_internal.h
 90 usr/src/lib/krb5/kadm5/kadm_err.h
 91 usr/src/lib/krb5/kadm5/kadm_rpc.h
 92 usr/src/lib/krb5/kadm5/server_internal.h
 93 usr/src/lib/krb5/kadm5/srv/server_acl.h
 94 usr/src/lib/krb5/kdb/adb_err.h
 95 usr/src/lib/krb5/kdb/kdb5.h
 96 usr/src/lib/krb5/plugins/kdb/db2/kdb_compat.h
 97 usr/src/lib/krb5/plugins/kdb/db2/kdb_db2.h
 98 usr/src/lib/krb5/plugins/kdb/db2/kdb_xdr.h
 99 usr/src/lib/krb5/plugins/kdb/db2/libdb2/btree/btree.h
100 usr/src/lib/krb5/plugins/kdb/db2/libdb2/btree/extern.h
101 usr/src/lib/krb5/plugins/kdb/db2/libdb2/hash/extern.h
102 usr/src/lib/krb5/plugins/kdb/db2/libdb2/hash/hash.h
103 usr/src/lib/krb5/plugins/kdb/db2/libdb2/hash/page.h
104 usr/src/lib/krb5/plugins/kdb/db2/libdb2/hash/search.h
105 usr/src/lib/krb5/plugins/kdb/db2/libdb2/include/db-int.h
106 usr/src/lib/krb5/plugins/kdb/db2/libdb2/include/db-ndbm.h
107 usr/src/lib/krb5/plugins/kdb/db2/libdb2/include/db-queue.h
108 usr/src/lib/krb5/plugins/kdb/db2/libdb2/mpool/mpool.h
109 usr/src/lib/krb5/plugins/kdb/db2/libdb2/recno/extern.h
110 usr/src/lib/krb5/plugins/kdb/db2/libdb2/recno/recno.h
111 usr/src/lib/krb5/plugins/kdb/db2/policy_db.h
112 usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/kdb_ldap.h
113 usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/kdb_xdr.h
114 usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_err.h
115 usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_handle.h
116 usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_krbcontainer.h
117 usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_main.h
118 usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_misc.h
119 usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_principal.h
120 usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_pwd_policy.h
121 usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_realm.h
122 usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_service_stash.h
123 usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_services.h
124 usr/src/lib/krb5/plugins/kdb/ldap/libkdb_ldap/ldap_tkt_policy.h
125 usr/src/lib/krb5/plugins/preauth/pkinit/pkinit_accessor.h
126 usr/src/lib/krb5/plugins/preauth/pkinit/pkinit_crypto_openssl.h
127 usr/src/lib/krb5/plugins/preauth/pkinit/pkinit_crypto.h
```

```
 128 usr/src/lib/krb5/plugins/preauth/pkinit/pkinit.h
 129 usr/src/lib/krb5/ss/copyright.h
 130 usr/src/lib/krb5/ss/mit-sipb-copyright.h
 131 usr/src/lib/krb5/ss/ss_internal.h
 132 usr/src/lib/krb5/ss/ss.h
 133 usr/src/lib/libc/port/locale/utils.h
 134 usr/src/lib/librstp/common/base.h
 135 usr/src/lib/librstp/common/choose.h
 136 usr/src/lib/librstp/common/edge.h
 137 usr/src/lib/librstp/common/migrate.h
 138 usr/src/lib/librstp/common/p2p.h
 139 usr/src/lib/librstp/common/pcost.h
 140 usr/src/lib/librstp/common/port.h
 141 usr/src/lib/librstp/common/portinfo.h
 142 usr/src/lib/librstp/common/rolesel.h
 143 usr/src/lib/librstp/common/roletrns.h
 144 usr/src/lib/librstp/common/statmch.h
 145 usr/src/lib/librstp/common/stp_bpdu.h
 146 usr/src/lib/librstp/common/stp_in.h
 147 usr/src/lib/librstp/common/stp_to.h
 148 usr/src/lib/librstp/common/stp_vectors.h
 149 usr/src/lib/librstp/common/stpm.h
 150 usr/src/lib/librstp/common/sttrans.h
 151 usr/src/lib/librstp/common/times.h
 152 usr/src/lib/librstp/common/topoch.h
 153 usr/src/lib/librstp/common/transmit.h
 154 usr/src/lib/librstp/common/uid_stp.h
 155 usr/src/lib/librstp/common/vector.h
 156 usr/src/uts/common/gssapi/mechs/krb5/include/aes_s2k.h
 157 usr/src/uts/common/gssapi/mechs/krb5/include/auth_con.h
 158 usr/src/uts/common/gssapi/mechs/krb5/include/cksumtypes.h
 159 usr/src/uts/common/gssapi/mechs/krb5/include/crc-32.h
 160 usr/src/uts/common/gssapi/mechs/krb5/include/des_int.h
 161 usr/src/uts/common/gssapi/mechs/krb5/include/dk.h
 162 usr/src/uts/common/gssapi/mechs/krb5/include/enc_provider.h
 163 usr/src/uts/common/gssapi/mechs/krb5/include/etypes.h
 164 usr/src/uts/common/gssapi/mechs/krb5/include/gssapi_generic.h
 165 usr/src/uts/common/gssapi/mechs/krb5/include/gssapi_krb5.h
 166 usr/src/uts/common/gssapi/mechs/krb5/include/gssapiP_generic.h
 167 usr/src/uts/common/gssapi/mechs/krb5/include/gssapiP_krb5.h
 168 usr/src/uts/common/gssapi/mechs/krb5/include/hash_provider.h
 169 usr/src/uts/common/gssapi/mechs/krb5/include/k5-int.h
 170 usr/src/uts/common/gssapi/mechs/krb5/include/k5-platform-load_16.h
 171 usr/src/uts/common/gssapi/mechs/krb5/include/k5-platform-load_32.h
 172 usr/src/uts/common/gssapi/mechs/krb5/include/k5-platform-load_64.h
 173 usr/src/uts/common/gssapi/mechs/krb5/include/k5-platform-store_16.h
 174 usr/src/uts/common/gssapi/mechs/krb5/include/k5-platform-store_32.h
 175 usr/src/uts/common/gssapi/mechs/krb5/include/k5-platform-store_64.h
 176 usr/src/uts/common/gssapi/mechs/krb5/include/k5-platform.h
 177 usr/src/uts/common/gssapi/mechs/krb5/include/k5-thread.h
 178 usr/src/uts/common/gssapi/mechs/krb5/include/keyhash_provider.h
 179 usr/src/uts/common/gssapi/mechs/krb5/include/krb5.h
 180 usr/src/uts/common/gssapi/mechs/krb5/include/old.h
 181 usr/src/uts/common/gssapi/mechs/krb5/include/raw.h
 182 usr/src/uts/common/gssapi/mechs/krb5/include/rsa-md4.h
 183 usr/src/uts/common/io/ixgbe/ixgbe_common.h
 184 usr/src/uts/intel/sys/acpi/acdebug.h
 185 usr/src/uts/intel/sys/acpi/acdisasm.h
 186 usr/src/uts/intel/sys/acpi/acevents.h
 187 usr/src/uts/intel/sys/acpi/acinterp.h
 188 usr/src/uts/intel/sys/acpi/acmacros.h
 189 usr/src/uts/intel/sys/acpi/acnames.h
 190 usr/src/uts/intel/sys/acpi/acpredef.h
 191 usr/src/uts/intel/sys/acpi/acresrc.h
 192 usr/src/uts/intel/sys/acpi/acstruct.h
 193 usr/src/uts/intel/sys/acpi/amlresrc.h
```



```
 194 usr/src/uts/intel/sys/acpi/platform/acwin64.h
```

```
**********************************************************
   18255 Sat Jul 19 14:19:57 2014
new/usr/src/cmd/man/makewhatis.c
Latest round of fixes per RM and AL.  Fix bugs found in man.c.
**********************************************************
```

```
/*
 * Copyright (c) 2002 John Rochester
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer,
 *    in this position and unchanged.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. The name of the author may not be used to endorse or promote products
 *    derived from this software without specific prior written permission
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ''AS IS'' AND ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
 * IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
 * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
 * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */

/*
 * Copyright 2012 Nexenta Systems, Inc. All rights reserved.
 * Copyright 2014 Garrett D'Amore <garrett@damore.org>
 */

#include <sys/types.h>
#include <sys/stat.h>
#include <sys/param.h>

#include <ctype.h>
#include <dirent.h>
#include <err.h>
#include <signal.h>
#include <stddef.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>

#include "man.h"
#include "stringlist.h"


/* Information collected about each man page in a section */
struct page_info {
        char    *filename;
        char    *name;
        char    *suffix;
        ino_t   inode;
};
```

_____*unchanged_portion_omitted_*

```
/*
```



```
 * Ensure that there is enough room in the sbuf
 * for nchars more characters.
 */
static void
sbuf_need(struct sbuf *sbuf, int nchars)
{
        char *new_content;
        size_t size, cntsize;
        size_t grow = 128;

        while (grow < nchars) {
                grow += 128;    /* we grow in chunks of 128 bytes */
        }

        /* Grow if the buffer isn't big enough */
        if (sbuf->end + nchars > sbuf->last) {
        /* Double the size of the allocation until the buffer is big enough */
        while (sbuf->end + nchars > sbuf->last) {
                size = sbuf->last + 1 - sbuf->content;
                size += grow;
                size *= 2;
                cntsize = sbuf->end - sbuf->content;

                if ((new_content = realloc(sbuf->content, size)) == NULL) {
                        perror("realloc");
                        if (tempfile[0] != '\0')
                                (void) unlink(tempfile);
                        exit(1);
                }
                new_content = (char *)malloc(size);
                (void) memcpy(new_content, sbuf->content, cntsize);
                free(sbuf->content);
                sbuf->content = new_content;
                sbuf->end = new_content + cntsize;
                sbuf->last = new_content + size - 1;
        }
}
```

_____*unchanged_portion_omitted_*

```
/*
 * Return true if no man page exists in the directory with
 * any of the names in the stringlist.
 */
static int
no_page_exists(char *dir, stringlist *names, char *suffix)
{
        char    path[MAXPATHLEN];
        char    *suffixes[] = { "", ".gz", ".bz2", NULL };
        size_t  i;
        int     j;

        for (i = 0; i < names->sl_cur; i++) {
                for (j = 0; suffixes[j] != NULL; j++) {
                        (void) snprintf(path, MAXPATHLEN, "%s/%s.%s%s",
                            dir, names->sl_str[i], suffix, suffixes[j]);
                        if (access(path, F_OK) == 0) {
                (void) snprintf(path, MAXPATHLEN, "%s/%s.%s.gz",
                    dir, names->sl_str[i], suffix);
                if (access(path, F_OK) < 0) {
                        path[strlen(path) - 3] = '\0';
                        if (access(path, F_OK) < 0)
                                continue;
                }
                                return (0);
                        }
                }
```

```
283         }
284         return (1);
285 }
```

*unchanged_portion_omitted_*

```
**********************************************************
   34083 Sat Jul 19 14:19:57 2014
new/usr/src/cmd/man/man.c
-T on the wrong command!
Latest round of fixes per RM and AL.  Fix bugs found in man.c.
**********************************************************
```

```
   1 /*
   2  * CDDL HEADER START
   3  *
   4  * The contents of this file are subject to the terms of the
   5  * Common Development and Distribution License (the "License").
   6  * You may not use this file except in compliance with the License.
   7  *
   8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9  * or http://www.opensolaris.org/os/licensing.
  10  * See the License for the specific language governing permissions
  11  * and limitations under the License.
  12  *
  13  * When distributing Covered Code, include this CDDL HEADER in each
  14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15  * If applicable, add the following below this CDDL HEADER, with the
  16  * fields enclosed by brackets "[]" replaced with your own identifying
  17  * information: Portions Copyright [yyyy] [name of copyright owner]
  18  *
  19  * CDDL HEADER END
  20  */

  22 /*
  23  * Copyright (c) 1990, 2010, Oracle and/or its affiliates. All rights reserved.
  24  * Copyright 2012, Josef 'Jeff' Sipek <jeffpc@31bits.net>. All rights reserved.
  25  * Copyright 2013 Nexenta Systems, Inc. All rights reserved.
  26  * Copyright 2014 Garrett D'Amore <garrett@damore.org>
  27  */

  29 /*      Copyright (c) 1983, 1984, 1985, 1986, 1987, 1988, 1989  AT&T.   */
  30 /*              All rights reserved.                                    */

  32 /*
  33  * University Copyright- Copyright (c) 1982, 1986, 1988
  34  * The Regents of the University of California
  35  * All Rights Reserved
  36  *
  37  * University Acknowledgment- Portions of this document are derived from
  38  * software developed by the University of California, Berkeley, and its
  39  * contributors.
  40  */

  42 /*
  43  * Find and display reference manual pages. This version includes makewhatis
  44  * functionality as well.
  45  */

  47 #include <sys/param.h>
  48 #include <sys/stat.h>
  49 #include <sys/termios.h>
  50 #include <sys/types.h>

  52 #include <ctype.h>
  53 #include <dirent.h>
  54 #include <err.h>
  55 #include <errno.h>
  56 #include <fcntl.h>
  57 #include <fnmatch.h>
  58 #include <limits.h>
  59 #include <locale.h>
  60 #include <malloc.h>
```



```
  61 #include <memory.h>
  62 #include <regex.h>
  63 #include <stdio.h>
  64 #include <stdlib.h>
  65 #include <string.h>
  66 #include <unistd.h>

  68 #include "man.h"


  71 /* Mapping of old directories to new directories */
  72 static const struct map_entry {
  73         char    *old_name;
  74         char    *new_name;
  75 } map[] = {
  76         { "3b",         "3ucb"          },
  77         { "3e",         "3elf"          },
  78         { "3g",         "3gen"          },
  79         { "3k",         "3kstat"        },
  80         { "3n",         "3socket"       },
  81         { "3r",         "3rt"           },
  82         { "3s",         "3c"            },
  83         { "3t",         "3thr"          },
  84         { "3x",         "3curses"       },
  85         { "3xc",        "3xcurses"      },
  86         { "3xn",        "3xnet"         },
  87         { NULL,         NULL            }
  85         { "3xn",        "3xnet"         }
  88 };
______unchanged_portion_omitted_

 154 static int      all = 0;
 155 static int      apropos = 0;
 156 static int      debug = 0;
 157 static int      found = 0;
 158 static int      list = 0;
 159 static int      makewhatis = 0;
 160 static int      printmp = 0;
 161 static int      sargs = 0;
 162 static int      psoutput = 0;
 163 static int      lintout = 0;
 164 static int      whatis = 0;
 165 static int      makewhatishere = 0;

 167 static char     *mansec;
 168 static char     *pager = NULL;
 165 static char     *pager;

 170 static char     *addlocale(char *);
 171 static struct man_node *build_manpath(char **, int);
 172 static void     do_makewhatis(struct man_node *);
 173 static char     *check_config(char *);
 174 static int      cmp(const void *, const void *);
 175 static int      dupcheck(struct man_node *, struct dupnode **);
 176 static int      format(char *, char *, char *, char *);
 177 static void     free_dupnode(struct dupnode *);
 178 static void     free_manp(struct man_node *manp);
 179 static void     freev(char **);
 180 static void     fullpaths(struct man_node **);
 181 static void     get_all_sect(struct man_node *);
 182 static int      getdirs(char *, char ***, int);
 183 static void     getpath(struct man_node *, char **);
 184 static void     getsect(struct man_node *, char **);
 185 static void     init_bintoman(void);
 186 static void     lower(char *);
 187 static void     mandir(char **, char *, char *, int);
```

```
188 static int      manual(struct man_node *, char *);
189 static char     *map_section(char *, char *);
190 static char     *path_to_manpath(char *);
191 static void     print_manpath(struct man_node *);
192 static void     search_whatis(char *, char *);
193 static int      searchdir(char *, char *, char *);
194 static void     sortdir(DIR *, char ***);
195 static char     **split(char *, char);
196 static void     usage_man(void);
197 static void     usage_whatapro(void);
198 static void     usage_catman(void);
199 static void     usage_makewhatis(void);
200 static void     whatapro(struct man_node *, char *);

202 static char     language[MAXPATHLEN];   /* LC_MESSAGES */
203 static char     localedir[MAXPATHLEN];  /* locale specific path component */

205 static char     *newsection = NULL;

207 static int      manwidth = 0;

209 extern const char       *__progname;

211 int
212 main(int argc, char **argv)
213 {
214         int             c, i;
215         char            **pathv;
216         char            *manpath = NULL;
217         static struct man_node *mandirs = NULL;
218         int             bmp_flags = 0;
219         int             ret = 0;
220         char            *opts;
221         char            *mwstr;
222         int             catman = 0;

224         (void) setlocale(LC_ALL, "");
225         (void) strcpy(language, setlocale(LC_MESSAGES, (char *)NULL));
226         if (strcmp("C", language) != 0)
227                 (void) strlcpy(localedir, language, MAXPATHLEN);

229 #if !defined(TEXT_DOMAIN)
230 #define TEXT_DOMAIN "SYS_TEST"
231 #endif
232         (void) textdomain(TEXT_DOMAIN);

234         if (strcmp(__progname, "apropos") == 0) {
235                 apropos++;
236                 opts = "M:ds:";
237         } else if (strcmp(__progname, "whatis") == 0) {
238                 apropos++;
239                 whatis++;
240                 opts = "M:ds:";
241         } else if (strcmp(__progname, "catman") == 0) {
242                 catman++;
243                 makewhatis++;
244                 opts = "P:M:w";
241                 opts = "M:w";
245         } else if (strcmp(__progname, "makewhatis") == 0) {
246                 makewhatis++;
247                 makewhatishere++;
248                 manpath = ".";
249                 opts = "";
250         } else {
251                 opts = "FM:P:T:adfklprs:tw";
252                 if (argc > 1 && strcmp(argv[1], "-") == 0) {
```



```
253                         pager = "cat";
254                         optind++;
248                 opts = "M:adfklps:tw";
255                 }
256         }

258         opterr = 0;
259         while ((c = getopt(argc, argv, opts)) != -1) {
260                 switch (c) {
261                 case 'M':       /* Respecify path for man pages */
262                         manpath = optarg;
263                         break;
264                 case 'a':
265                         all++;
266                         break;
267                 case 'd':
268                         debug++;
269                         break;
270                 case 'f':
271                         whatis++;
272                         /*FALLTHROUGH*/
273                 case 'k':
274                         apropos++;
275                         break;
276                 case 'l':
277                         list++;
278                         all++;
279                         break;
271                         /*FALLTHROUGH*/
280                 case 'p':
281                         printmp++;
282                         break;
283                 case 's':
284                         mansec = optarg;
285                         sargs++;
286                         break;
287                 case 'r':
288                         lintout++;
289                         break;
290                 case 't':
291                         psoutput++;
292                         break;
293                 case 'T':
294                 case 'P':
295                 case 'F':
296                         /* legacy options, compatibility only and ignored */
297                         break;
298                 case 'w':
299                         makewhatis++;
300                         break;
301                 case '?':
302                 default:
303                         if (apropos)
304                                 usage_whatapro();
305                         else if (catman)
306                                 usage_catman();
307                         else if (makewhatishere)
308                                 usage_makewhatis();
309                         else
310                                 usage_man();
311                 }
312         }
313         argc -= optind;
314         argv += optind;

316         if (argc == 0) {
```

```
317                 if (apropos) {
318                         (void) fprintf(stderr, gettext("%s what?\n"),
319                             __progname);
320                         exit(1);
321                 } else if (!printmp && !makewhatis) {
322                         (void) fprintf(stderr,
323                             gettext("What manual page do you want?\n"));
324                         exit(1);
325                 }
326         }

328         init_bintoman();
329         if (manpath == NULL && (manpath = getenv("MANPATH")) == NULL) {
330                 if ((manpath = getenv("PATH")) != NULL)
331                         bmp_flags = BMP_ISPATH | BMP_APPEND_DEFMANDIR;
332                 else
333                         manpath = DEFMANDIR;
334         }
335         pathv = split(manpath, ':');
336         mandirs = build_manpath(pathv, bmp_flags);
337         freev(pathv);
338         fullpaths(&mandirs);

340         if (makewhatis) {
341                 do_makewhatis(mandirs);
342                 exit(0);
343         }

345         if (printmp) {
346                 print_manpath(mandirs);
347                 exit(0);
348         }

350         /* Collect environment information */
351         if (isatty(STDOUT_FILENO) && (mwstr = getenv("MANWIDTH")) != NULL &&
352             *mwstr != '\0') {
353                 if (strcasecmp(mwstr, "tty") == 0) {
354                         struct winsize  ws;

356                         if (ioctl(0, TIOCGWINSZ, &ws) != 0)
357                                 warn("TIOCGWINSZ");
358                         else
359                                 manwidth = ws.ws_col;
360                 } else {
361                         manwidth = (int)strtol(mwstr, (char **)NULL, 10);
362                         if (manwidth < 0)
363                                 manwidth = 0;
364                 }
365         }
366         if (manwidth != 0) {
367                 DPRINTF("-- Using non-standard page width: %d\n", manwidth);
368         }

370         if (pager == NULL) {
371                 if ((pager = getenv("PAGER")) == NULL || *pager == '\0')
372                         pager = PAGER;
373         }
374         DPRINTF("-- Using pager: %s\n", pager);

376         for (i = 0; i < argc; i++) {
377                 char            *cmd;
378                 static struct man_node *mp;
379                 char            *pv[2];

381                 /*
382                  * If full path to command specified, customize
```



```
383                  * the manpath accordingly.
384                  */
385                 if ((cmd = strrchr(argv[i], '/')) != NULL) {
386                         *cmd = '\0';
387                         if ((pv[0] = strdup(argv[i])) == NULL)
388                                 err(1, "strdup");
389                         pv[1] = NULL;
390                         *cmd = '/';
391                         mp = build_manpath(pv,
392                             BMP_ISPATH | BMP_FALLBACK_DEFMANDIR);
393                 } else {
394                         mp = mandirs;
395                 }

397                 if (apropos)
398                         whatapro(mp, argv[i]);
399                 else
400                         ret += manual(mp, argv[i]);

402                 if (mp != NULL && mp != mandirs) {
403                         free(pv[0]);
404                         free_manp(mp);
405                 }
406         }

408         return (ret == 0 ? 0 : 1);
409 }
```

*unchanged_portion_omitted_*

```
1159 /*
1160  * Check the hash table of old directory names to see if there is a
1161  * new directory name.
1162  */
1163 static char *
1164 map_section(char *section, char *path)
1165 {
1166         int     i;
1149         int     len;
1167         char    fullpath[MAXPATHLEN];

1169         if (list)  /* -l option fall through */
1170                 return (NULL);

1172         for (i = 0; map[i].new_name != NULL; i++) {
1173                 if (strcmp(section, map[i].old_name) == 0) {
1155         for (i = 0; i <= ((sizeof (map)/sizeof (map[0]) - 1)); i++) {
1156                 if (strlen(section) > strlen(map[i].new_name)) {
1157                         len = strlen(section);
1158                 } else {
1159                         len = strlen(map[i].new_name);
1160                 }
1161                 if (strncmp(section, map[i].old_name, len) == 0) {
1174                         (void) snprintf(fullpath, sizeof (fullpath),
1175                             "%s/man%s", path, map[i].new_name);
1163                         "%s/sman%s", path, map[i].new_name);
1176                         if (!access(fullpath, R_OK | X_OK)) {
1177                                 return (map[i].new_name);
1178                         } else {
1179                                 return (NULL);
1180                         }
1181                 }
1182         }

1184         return (NULL);
1185 }
```

```
1187 /*
1188  * Format the manpage.
1189  */
1190 static int
1191 format(char *path, char *dir, char *name, char *pg)
1192 {
1193         char            manpname[MAXPATHLEN], catpname[MAXPATHLEN];
1194         char            cmdbuf[BUFSIZ], tmpbuf[BUFSIZ];
1195         char            *cattool;
1196         int             utf8 = 0;
1197         struct stat     sbman, sbcat;

1199         found++;

1201         if (list) {
1202                 (void) printf(gettext("%s(%s)\t-M %s\n"), name, dir + 3, path);
1203                 return (-1);
1204         }

1206         (void) snprintf(manpname, sizeof (manpname), "%s/man%s/%s", path,
1207             dir + 3, pg);
1208         (void) snprintf(catpname, sizeof (catpname), "%s/cat%s/%s", path,
1209             dir + 3, pg);

1211         /* Can't do PS output if manpage doesn't exist */
1212         if (stat(manpname, &sbman) != 0 && (psoutput|lintout))
1200         if (stat(manpname, &sbman) != 0 && psoutput)
1213                 return (-1);

1215         /*
1216          * If both manpage and catpage do not exist, manpname is
1217          * broken symlink, most likely.
1218          */
1219         if (stat(catpname, &sbcat) != 0 && stat(manpname, &sbman) != 0)
1220                 err(1, "%s", manpname);

1222         /* Setup cattool */
1223         if (fnmatch("*.gz", manpname, 0) == 0)
1224                 cattool = "gzcat";
1225         else if (fnmatch("*.bz2", manpname, 0) == 0)
1226                 cattool = "bzcat";
1227         else
1228                 cattool = "cat";
1216                 cattool = "gzcat -f";

1230         /* Preprocess UTF-8 input with preconv (could be smarter) */
1231         if (strstr(path, "UTF-8") != NULL)
1232                 utf8 = 1;

1234         if (psoutput) {
1235                 (void) snprintf(cmdbuf, BUFSIZ,
1236                     "cd %s; %s %s%s | mandoc -Tps | lp -Tpostscript",
1237                     path, cattool, manpname,
1238                     utf8 ? " | " PRECONV " -e UTF-8" : "");
1239                 DPRINTF("-- Using manpage: %s\n", manpname);
1240                 goto cmd;
1241         } else if (lintout) {
1242                 (void) snprintf(cmdbuf, BUFSIZ,
1243                     "cd %s; %s %s%s | mandoc -Tlint",
1244                     path, cattool, manpname,
1245                     utf8 ? " | " PRECONV " -e UTF-8" : "");
1246                 DPRINTF("-- Linting manpage: %s\n", manpname);
1247                 goto cmd;
1248         }

1250         /*
```



```
1251          * Output catpage if:
1252          * - manpage doesn't exist
1253          * - output width is standard and catpage is recent enough
1254          */
1255         if (stat(manpname, &sbman) != 0 || (manwidth == 0 &&
1256             stat(catpname, &sbcat) == 0 && sbcat.st_mtime >= sbman.st_mtime)) {
1257                 DPRINTF("-- Using catpage: %s\n", catpname);
1258                 (void) snprintf(cmdbuf, BUFSIZ, "%s %s", pager, catpname);
1259                 goto cmd;
1260         }

1262         DPRINTF("-- Using manpage: %s\n", manpname);
1263         if (manwidth > 0)
1264                 (void) snprintf(tmpbuf, BUFSIZ, "-Owidth=%d ", manwidth);
1265         (void) snprintf(cmdbuf, BUFSIZ, "cd %s; %s %s%s | mandoc -T%s %s| %s",
1266             path, cattool, manpname,
1267             utf8 ? " | " PRECONV " -e UTF-8 " : "",
1268             utf8 ? "utf8" : "ascii", (manwidth > 0) ? tmpbuf : "", pager);

1270 cmd:
1271         DPRINTF("-- Command: %s\n", cmdbuf);

1273         if (!debug)
1274                 return (system(cmdbuf) == 0);
1275         else
1276                 return (0);
1277 }
______unchanged_portion_omitted_

1584 static void
1585 usage_man(void)
1586 {

1588         (void) fprintf(stderr, gettext(
1589 "usage: man [-alptw] [-M path] [-s section] name ...\n"
1590 "       man [-M path] [-s section] -k keyword ...\n"
1591 "       man [-M path] [-s section] -f keyword ...\n"));
1571 "       man [-M path] [-s section] -k keyword -- emulate apropos\n"
1572 "       man [-M path] [-s section] -f keyword -- emulate whatis\n"));

1593         exit(1);
1594 }
______unchanged_portion_omitted_
```

```
**********************************************************
    1745 Sat Jul 19 14:19:58 2014
new/usr/src/man/man1/apropos.1
Latest round of fixes per RM and AL.  Fix bugs found in man.c.
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2012 Nexenta Systems, Inc. All rights reserved.
.\" Copyright 2014 Garrett D'Amore <garrett@damore.org>
.\"
.Dd Jul 18, 2014
.Dd Jul 13, 2014
.Dt APROPOS 1
.Os
.Sh NAME
.Nm apropos ,
.Nm whatis
.Nd keyword search in
.Nm whatis
database files
.Sh SYNOPSIS
.Nm
.Op Fl M Ar path
.Op Fl s Ar section
.Ar keyword ...
.Nm whatis
.Op Fl M Ar path
.Op Fl s Ar section
.Ar keyword ...
.Sh DESCRIPTION
The
.Nm
utility searches a set of
.Nm whatis
database files matching each
.Ar keyword .
The
.Nm whatis
utility does the same search but only on complete words. The
.Nm whatis
database files are created using the
database files are created using
.Xr man 1
command.
.Sh OPTIONS
.Bl -tag -width ".Fl d"
.It Fl M Ar path
Force a specific colon separated manual path instead of the default search path.
Overrides the
.Ev MANPATH
environment variable.
.It Fl s Ar section
Restrict search to specified
.Ar section .
.El
.Sh ENVIRONMENT
The following environment variables affect the execution of
```



```
.Nm :
.Bl -tag -width ".Ev MANPATH , PATH"
.It Ev MANPATH , PATH
Used to find the location of the
.Nm whatis
database files.
.El
.Sh DIAGNOSTICS
The
.Nm
utility exits 0 if a keyword matched and 1 if no keywords are matched or no
.Nm whatis
databases are found.
.Sh INTERFACE STABILITY
.Nm Committed .
.Sh CODE SET INDEPENDENCE
Enabled.
.Sh SEE ALSO
.Xr man 1 ,
.Xr mandoc 1
```

```
**********************************************************
    9374 Sat Jul 19 14:19:58 2014
new/usr/src/man/man1/man.1
Latest round of fixes per RM and AL.  Fix bugs found in man.c.
**********************************************************
```

```
.\" Copyright 2014 Garrett D'Amore <garrett@damore.o
.\" Copyright (c) 2008, Sun Microsystems, Inc. All Rights Reserved.
.\" Copyright (c) 1980 Regents of the University of California.
.\" The Berkeley software License Agreement specifies the terms and conditions
.\" for redistribution.
.Dd Jul 18, 2014
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright 2012 Nexenta Systems, Inc. All rights reserved.
.\" Copyright 2014 Garrett D'Amore <garrett@damore.org>
.\"
.Dd October 18, 2012
.Dt MAN 1
.Os
.Sh NAME
.Nm man
.Nd find and display reference manual pages
.Sh SYNOPSIS
.Nm
.Op Fl
.Op Fl adFlrt
.Op Fl T Ar macro-package
.Op Fl M Ar path
.Op Fl s Ar section
.Ar name ...
.Op Fl alptw
.Op Fl M Ar manpath
.Op Fl s Ar mansect
.Ar page ...
.Nm
.Op Fl M Ar path
.Op Fl s Ar section
.Fl k
.Ar keyword
.Ar ...
.Nm
.Op Fl M Ar path
.Op Fl s Ar section
.Op Fl s Ar mansect
.Fl f
.Ar
.Ar keyword ...
.Nm
.Op Fl M Ar path
.Fl w
.Op Fl s Ar mansect
.Fl k
.Ar keyword ...
.Sh DESCRIPTION
The
.Nm
command displays information from the reference manuals. It
```



```
displays complete manual pages that you select by
.Ar name ,
or one-line summaries selected either by
.Ar keyword
.Pq Fl k ,
or by the name of an associated file
.Pq Fl f .
If no manual page is located,
.Nm
prints an error message.
.Ss "Source Format"
Reference Manual pages are marked up with either
.Xr man 5 ,
or
.Xr mdoc 5
language tags. The
.Nm
command recognizes the type of markup and
processes the file accordingly.
.
.Ss "Location of Manual Pages"
.
The online Reference Manual page directories are conventionally located in
.Pa /usr/share/man .
Each directory corresponds to a
section of the manual. Since these directories are optionally installed, they
might not reside on your host. You might have to mount
.Pa /usr/share/man
from a host on which they do reside.
The
.Nm
command reformats a page whenever it is requested.
utility finds and displays reference manual pages.
.Pp
If the standard output is not a terminal, or if the
.Fl
flag is given,
Options that
.Nm
pipes its output through
.Xr cat 1 .
Otherwise,
.Nm
pipes its output through a pager such as
.Xr more 1
to handle paging and underlining on the screen.
.Sh OPTIONS
The following options are supported:
understands:
.Bl -tag -width indent
.It Fl a
Shows all manual pages matching
.Ar name
within the
.It Fl M Ar manpath
Forces a specific colon separated manual path instead of the default
search path.
Overrides the
.Ev MANPATH
search path. Manual pages are displayed in the order found.
.It Fl d
Debugs. Displays what a section-specifier evaluates to, method used for
searching, and paths searched by
.Nm .
.It Fl f Ar file ...
Attempts to locate manual pages related to any of the given
```

```
 97 .Ar file
 98 names. It strips the leading path name components from each
 99 .Ar file ,
100 and then prints one-line summaries containing the resulting basename or names.
101 This option also uses the
102 .Pa whatis
103 database.
104 .It Fl F
105 This option is present for backwards compatibility and is documented
106 here for reference only.  It performs no function.
107 .It Fl k Ar keyword ...
108 Prints out one-line summaries from the
109 .Pa whatis
110 database (table of contents) that contain any of the given
111 .Ar keyword .
112 The
113 .Pa whatis
114 database is created using the
115 .Fl w
116 option.
117 .It Fl l
118 Lists all manual pages found matching
119 .Ar name
120 within the search path.
121 .It Fl M Ar path
122 Specifies an alternate search path for manual pages. The
123 .Ar path
124 is a colon-separated list of directories that contain manual page directory
125 subtrees. For example, if
126 .Ar path
127 is
128 .Pa /usr/share/man:/usr/local/man ,
129 .Nm
130 searches for
131 .Ar name
132 in the standard location, and then
133 .Pa /usr/local/man .
134 When used with the
135 .Fl k ,
136 .Fl f ,
137 or
138 .Fl w
139 options, the
140 .Fl M
141 option must appear first. Each directory in the
142 .Ar path
143 is assumed to contain subdirectories of the form
144 .Pa man* ,
145 one for each section. This option overrides the
146 .Ev MANPATH
147 environment variable.
148 .It Fl r
149 Reformats the manual page, checking for formatting errors, but does not
150 display it.
151 .It Fl s Ar section
152 Specifies sections of the manual for
153 .Nm
154 to search. The directories searched for
155 .Ar name
156 are limited to those specified by
157 .Ar section .
158 .Ar section
159 can be a numerical digit, perhaps followed by one or more letters
160 to match the desired section of the manual, for example,
161 .Li "3libucb".
162 Also,
```



```
163 .Ar section
164 can be a word, for example,
165 .Li local ,
166 .Li new ,
167 .Li old ,
168 .Li public .
169 .Ar section
170 can also be a letter. To specify multiple sections,
171 separate each section with a comma. This option overrides the
172 .Ev MANPATH
173 environment variable and the
174 .Pa man.cf
175 file. See
176 .Sx Search Path
177 below for an explanation of how
178 .Nm
179 conducts its search.
 50 .It Fl a
 51 Display all manual pages instead of just the first found for each
 52 .Ar page
 53 argument.
 54 .It Fl f
 55 Emulate
 56 .Xr whatis 1 .
 57 .It Fl k
 58 Emulate
 59 .Xr apropos 1 .
 60 .It Fl l
 61 Display the location of the manual page instead of the contents of
 62 the manual page.
 63 .It Fl p
 64 Output current path used for searching.
 65 .It Fl s Ar mansect
 66 Restrict manual sections searched to the specified colon delimited list.
180 .It Fl t
181 Arranges for the specified manual pages to be sent to the default
182 printer as PostScript.
183 .It Fl T Ar macro-package
184 This option is present for backwards compatibility and is documented
185 here for reference only.  It performs no function.
 68 Send the content formatted as PostScript to the default printer.
186 .It Fl w
187 Updates the
 70 Create
188 .Nm whatis
189 database.
190 .El
191 .Sh OPERANDS
192 The following operand is supported:
193 .Bl -tag -width indent
194 .It Ar name
195 The name of a standard utility or a keyword.
196 .El
197 .Sh USAGE
198 The usage of
199 .Nm
200 is described below:
201 .
202 .Ss "Manual Page Sections"
203 .
204 Entries in the reference manuals are organized into
205 .Em sections .
206 A section
207 name consists of a major section name, typically a single digit, optionally
208 followed by a subsection name, typically one or more letters. An unadorned
209 major section name, for example,
```

```
.Qq 9 ,
does not act as an abbreviation for
the subsections of that name, such as
.Qq 9e ,
.Qq 9f ,
or
.Qq 9s .
That is, each subsection must be searched separately by
.Nm
.Fl s .
Each section contains descriptions apropos to a particular reference category,
with subsections refining these distinctions. See the
.Em intro
manual pages for an explanation of the classification used in this release.
.
.Ss "Search Path"
.
Before searching for a given
.Ar name ,
.Nm
constructs a list of candidate directories and sections.
It searches for
.Ar name
in the directories specified by the
.Ev MANPATH
environment variable.
.Lp
In the absence of
.Ev MANPATH ,
.Nm
constructs its search path based upon the
.Ev PATH
environment variable, primarily by substituting
.Li man
for the last component of the
.Ev PATH
element. Special provisions are added
to account for unique characteristics of directories such as
.Pa /sbin ,
.Pa /usr/ucb ,
.Pa /usr/xpg4/bin ,
and others. If the file argument contains
a
.Qq /
character, the
.Em dirname
portion of the argument is used in place of
.Ev PATH
elements to construct the search path.
.Lp
Within the manual page directories,
.Nm
confines its search to the
sections specified in the following order:
.Bl -bullet
.It
.Ar sections
specified on the command line with the
.Fl s
option
.It
.Ar sections
embedded in the
.Ev MANPATH
environment variable
.It
```



```
.Ar sections
specified in the
.Pa man.cf
file for each directory specified in the
.Ev MANPATH
environment variable
.El
If none of the above exist,
.Nm
searches each directory in the manual
page path, and displays the first matching manual page found.
.Lp
The
.Pa man.cf
file has the following format:
.Lp
.Dl Pf MANSECTS= Ar section , Ns Op Ar section...
.Lp
Lines beginning with
.Sq Li #
and blank lines are considered comments, and are
ignored. Each directory specified in
.Ev MANPATH
can contain a manual page
configuration file, specifying the default search order for that directory.
.Sh "Referring to Other Manual Pages"
If the first line of the manual page is a reference to another manual
page entry fitting the pattern:
.Lp
.Dl \&.so man*/\fIsourcefile\fR
.Lp
.Nm
processes the indicated file in place of the current one. The
reference must be expressed as a path name relative to the root of the manual
page directory subtree.
.Lp
When the second or any subsequent line starts with \fB\&.so\fR, \fBman\fR
ignores it; \fBtroff\fR(1) or \fBnroff\fR(1) processes the request in the usual
manner.
.Sh ENVIRONMENT VARIABLES
See
.Xr environ 5
for descriptions of the following environment variables
that affect the execution of
.Nm man :
.Ev LANG ,
.Ev LC_ALL ,
.Ev LC_CTYPE ,
.Ev LC_MESSAGES ,
 72 databases used by
 73 .Xr apropos 1
and
.Ev NLSPATH .
.Bl -tag -width indent
 75 .Xr whatis 1 .
 76 .El
 77 .Sh ENVIRONMENT
 78 The following environment variables affect the execution of
 79 .Nm :
 80 .Bl -tag -width ".Ev MANPATH"
 81 .It Ev LC_ALL , LC_CTYPE , LANG
 82 Used to find locale-specific manual pages.
.It Ev MANPATH
A colon-separated list of directories; each directory can be followed by a
comma-separated list of sections. If set, its value overrides
\fB/usr/share/man\fR as the default directory search path, and the \fBman.cf\fR
```

```
file as the default section search path. The \fB-M\fR and \fB-s\fR flags, in
turn, override these values.)
Used to find the location of the manual files.
Corresponds to the
.Fl M
option.
.It Ev MANWIDTH
Defines the width of output. If set to
.Dq Li tty ,
and output is to a terminal, full width of terminal is used.
.It Ev PAGER
A program to use for interactively delivering
output to the screen. If not set,
.Sq Nm more Fl s
is used. See
.Xr more 1 .
Program used to display files. If unset,
.Dq Li "less -ins"
is used.
.It Ev PATH
Used to find location of manual files if
.Ev MANPATH
and
.Fl M
are not specified.
.El
.Sh FILES
.Bl -tag -width indent
.It Pa /usr/share/man
Root of the standard manual page directory subtree
.It Pa /usr/share/man/man?/*
Unformatted manual entries
.It Pa /usr/share/man/whatis
Table of contents and keyword database
.Bl -tag -width indent -compact
.It Pa man.cf
Default search order by section
.El
.Sh EXIT STATUS
.Ex -std man
.Sh EXAMPLES
.
.Ss Example 1: Creating a PostScript Version of a man page
.
The following example spools the
.Xr pipe 2
man page in PostScript to the default printer:
Per-manpath configuration settings.  The file is formatted as follows:
.Bd -literal -offset indent
MANSECT=\fIsection\fR[,\fIsection\fR]...
.Ed
.Pp
.Dl % man -t -s 2 pipe
.Pp
Note that
.Xr mandoc 1
can be used to obtain the PostScript content directly.
.Ss Example 2: Creating a Text Version of a man page
The following example creates the
.Xr pipe 2
man page in ASCII text:
.Pp
.Dl % man pipe.2 | col -x -b > pipe.text
Each section consists of a section in the reference manual.  The file
may also contain comment blank lines or lines consisting of comments, where
the first character in the line is '#'.  Both blank lines and comment lines are
```



```
ignored.
.El
.Sh CODE SET INDEPENDENCE
Enabled.
.Sh INTERFACE STABILITY
The
.Nm
utility is
.Nm Standard ,
as is the
.Fl k
option. The other options are
.Nm Committed .
.Sh SEE ALSO
.Xr apropos 1 ,
.Xr cat 1 ,
.Xr col 1 ,
.Xr intro 1 ,
.Xr mandoc 1 ,
.Xr more 1 ,
.Xr whatis 1 ,
.Xr environ 5 ,
.Xr man 5 ,
.Xr mdoc 5
.Xr mdoc 5 ,
.Xr standards 5
.Sh NOTES
The
.Fl f
and
.Fl k
options use the
.Nm whatis
database, which is
created with the
.Fl w
option.
.Sh BUGS
The manual is supported to be reproducible either on a phototypesetter or on an
ASCII terminal. However, on a terminal some information (indicated by
font changes, for instance) is lost.
Some pages may contain information which cannot be properly displayed on
all terminals.  In such cases, some information may be lost.
```

```
**********************************************************
    1590 Sat Jul 19 14:19:58 2014
new/usr/src/man/man1m/catman.1m
Latest round of fixes per RM and AL.  Fix bugs found in man.c.
**********************************************************
```

```
   1 .\"
   2 .\" This file and its contents are supplied under the terms of the
   3 .\" Common Development and Distribution License ("CDDL"), version 1.0.
   4 .\" You may only use this file in accordance with the terms of version
   5 .\" 1.0 of the CDDL.
   6 .\"
   7 .\" A full copy of the text of the CDDL should have accompanied this
   8 .\" source.  A copy of the CDDL is also available via the Internet at
   9 .\" http://www.illumos.org/license/CDDL.
  10 .\"
  11 .\"
  12 .\" Copyright 2014 Garrett D'Amore <garrett@damore.org>
  13 .\"
  14 .Dd Jul 19, 2014
  15 .Dt CATMAN 1M
  14 .Dd Jul 16, 2014
  15 .Dt CATAMN 1M
  16 .Os
  17 .Sh NAME
  18 .Nm catman
  19 .Nd generate
  20 .Nm whatis
  21 database files
  22 .Sh SYNOPSIS
  23 .Nm
  24 .Op Fl M Ar path
  25 .Op Fl w
  26 .Sh DESCRIPTION
  27 The
  28 .Nm
  29 utility generates a set of
  30 .Nm whatis
  31 database files suitable for use with
  32 .Xr apropos 1
  33 and
  34 .Xr whatis 1 .
  35 It is supplied for compatibility reasons.  The same databases can
  36 be generated using the
  37 .Fl w
  38 option with
  39 .Xr man 1 ,
  40 and that command should be used instead.
  41 .Sh OPTIONS
  42 .Bl -tag -width ".Fl d"
  43 .It Fl M Ar path
  44 Generate the
  45 .Nm whatis
  46 database files within the specified colon separated manual paths.
  47 Overrides the
  48 .Ev MANPATH
  49 environment variable.
  50 .It Fl w
  51 This option is present for backwards compatibility, and is ignored.
  52 .El
  53 .Sh ENVIRONMENT
  54 The following environment variables affect the execution of
  55 .Nm :
  56 .Bl -tag -width ".Ev MANPATH"
  57 .It Ev MANPATH
  58 Used to specify a colon seperated list of manual paths within
  59 which to generate
```



```
  60 .Nm whatis
  61 database files.
  62 .El
  63 .Sh EXIT STATUS
  64 .Ex -std
  59 .Sh DIAGNOSTICS
  60 The
  61 .Nm
  62 utility exits 0 on success, and non-zero otherwise.
  65 .Sh INTERFACE STABILITY
  66 .Nm "Obsolete Committed" .
  67 .Sh CODE SET INDEPENDENCE
  68 Enabled.
  69 .Sh SEE ALSO
  70 .Xr apropos 1 ,
  71 .Xr man 1 ,
  72 .Xr whatis 1
```

```
**********************************************************
    7225 Sat Jul 19 14:19:58 2014
new/usr/src/man/man5/eqn.5
Latest round of fixes per RM and AL.  Fix bugs found in man.c.
**********************************************************
.\"
.\" Permission to use, copy, modify, and distribute this software for any
.\" purpose with or without fee is hereby granted, provided that the above
.\" copyright notice and this permission notice appear in all copies.
.\"
.\" THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
.\" WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
.\" MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
.\" ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
.\" WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
.\" ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
.\" OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
.\"
.\"
.\" Copyright (c) 2011 Kristaps Dzonsons <kristaps@bsd.lv>
.\" Copyright 2012 Nexenta Systems, Inc. All rights reserved.
.\"
.Dd Jul 19, 2014
.Dd Sep 25, 2011
.Dt EQN 5
.Os
.Sh NAME
.Nm eqn
.Nd eqn language reference for mandoc
.Sh DESCRIPTION
The
.Nm eqn
language is an equation-formatting language.
It is used within
.Xr mdoc 5
and
.Xr man 5
.Ux
manual pages.
It describes the
.Em structure
of an equation, not its mathematical meaning.
This manual describes the
.Nm
language accepted by the
.Xr mandoc 1
utility, which corresponds to the Second Edition eqn specification (see
.Sx SEE ALSO
for references).
.Pp
Equations within
.Xr mdoc 5
or
.Xr man 5
documents are enclosed by the standalone
.Sq \&.EQ
and
.Sq \&.EN
tags.
Equations are multi-line blocks consisting of formulas and control
statements.
.Sh EQUATION STRUCTURE
Each equation is bracketed by
.Sq \&.EQ
and
.Sq \&.EN
```



```
strings.
.Em Note :
these are not the same as
.Xr roff 5
macros, and may only be invoked as
.Sq \&.EQ .
.Pp
The equation grammar is as follows, where quoted strings are
case-sensitive literals in the input:
.Bd -literal -offset indent
eqn     : box | eqn box
box     : text
        | \*q{\*q eqn \*q}\*q
        | \*qdefine\*q text text
        | \*qndefine\*q text text
        | \*qtdefine\*q text text
        | \*qgfont\*q text
        | \*qgsize\*q text
        | \*qset\*q text text
        | \*qundef\*q text
        | box pos box
        | box mark
        | \*qmatrix\*q \*q{\*q [col \*q{\*q list \*q}\*q ]*
        | pile \*q{\*q list \*q}\*q
        | font box
        | \*qsize\*q text box
        | \*qleft\*q text eqn [\*qright\*q text]
col     : \*qlcol\*q | \*qrcol\*q | \*qccol\*q | \*qcol\*q
text    : [^space\e\*q]+ | \e\*q.*\e\*q
pile    : \*qlpile\*q | \*qcpile\*q | \*qrpile\*q | \*qpile\*q
pos     : \*qover\*q | \*qsup\*q | \*qsub\*q | \*qto\*q | \*qfrom\*q
mark    : \*qdot\*q | \*qdotdot\*q | \*qhat\*q | \*qtilde\*q | \*qvec\*q
        | \*qdyad\*q | \*qbar\*q | \*qunder\*q
font    : \*qroman\*q | \*qitalic\*q | \*qbold\*q | \*qfat\*q
list    : eqn
        | list \*qabove\*q eqn
space   : [\e^~ \et]
.Ed
.Pp
White-space consists of the space, tab, circumflex, and tilde
characters.
If within a quoted string, these space characters are retained.
Quoted strings are also not scanned for replacement definitions.
.Pp
The following text terms are translated into a rendered glyph, if
available: alpha, beta, chi, delta, epsilon, eta, gamma, iota, kappa,
lambda, mu, nu, omega, omicron, phi, pi, psi, rho, sigma, tau, theta,
upsilon, xi, zeta, DELTA, GAMMA, LAMBDA, OMEGA, PHI, PI, PSI, SIGMA,
THETA, UPSILON, XI, inter (intersection), union (union), prod (product),
int (integral), sum (summation), grad (gradient), del (vector
differential), times (multiply), cdot (centre-dot), nothing (zero-width
space), approx (approximately equals), prime (prime), half (one-half),
partial (partial differential), inf (infinity), >> (much greater), <<
(much less), \-> (left arrow), <\- (right arrow), += (plus-minus), !=
(not equal), == (equivalence), <= (less-than-equal), and >=
(more-than-equal).
.Pp
The following control statements are available:
.Bl -tag -width Ds
.It Cm define
Replace all occurrences of a key with a value.
Its syntax is as follows:
.Pp
.D1 define Ar key cvalc
.Pp
The first character of the value string,
```

```
.Ar c ,
is used as the delimiter for the value
.Ar val .
This allows for arbitrary enclosure of terms (not just quotes), such as
.Pp
.D1 define Ar foo 'bar baz'
.D1 define Ar foo cbar bazc
.Pp
It is an error to have an empty
.Ar key
or
.Ar val .
Note that a quoted
.Ar key
causes errors in some
.Nm
implementations and should not be considered portable.
It is not expanded for replacements.
Definitions may refer to other definitions; these are evaluated
recursively when text replacement occurs and not when the definition is
created.
.Pp
Definitions can create arbitrary strings, for example, the following is
a legal construction.
.Bd -literal -offset indent
define foo 'define'
foo bar 'baz'
.Ed
.Pp
Self-referencing definitions will raise an error.
The
.Cm ndefine
statement is a synonym for
.Cm define ,
while
.Cm tdefine
is discarded.
.It Cm gfont
Set the default font of subsequent output.
Its syntax is as follows:
.Pp
.D1 gfont Ar font
.Pp
In
.Xr mandoc 1 ,
this value is discarded.
In mandoc, this value is discarded.
.It Cm gsize
Set the default size of subsequent output.
Its syntax is as follows:
.Pp
.D1 gsize Ar size
.Pp
The
.Ar size
value should be an integer.
.It Cm set
Set an equation mode.
In
.Xr mandoc 1 ,
both arguments are thrown away.
In mandoc, both arguments are thrown away.
Its syntax is as follows:
.Pp
.D1 set Ar key val
.Pp
```



```
The
.Ar key
and
.Ar val
are not expanded for replacements.
This statement is a GNU extension.
.It Cm undef
Unset a previously-defined key.
Its syntax is as follows:
.Pp
.D1 define Ar key
.Pp
Once invoked, the definition for
.Ar key
is discarded.
The
.Ar key
is not expanded for replacements.
This statement is a GNU extension.
.El
.Sh COMPATIBILITY
This section documents the compatibility of
.Xr mandoc 1
This section documents the compatibility of mandoc
.Nm
and the
.Xr troff 1
and the troff
.Nm
implementation (including GNU troff).
.Pp
.Bl -dash -compact
.It
The text string
.Sq \e\*q
is interpreted as a literal quote in
.Xr troff 1 .
In
.Xr mandoc 1 ,
this is interpreted as a comment.
is interpreted as a literal quote in troff.
In mandoc, this is interpreted as a comment.
.It
In
.Xr troff 1 ,
The circumflex and tilde white-space symbols map to
In troff, The circumflex and tilde white-space symbols map to
fixed-width spaces.
In
.Xr mandoc 1 ,
these characters are synonyms for the space character.
In mandoc, these characters are synonyms for the space character.
.It
The
.Xr troff 1 ,
implementation of
The troff implementation of
.Nm
allows for equation alignment with the
.Cm mark
and
.Cm lineup
tokens.
.Xr mandoc 1
discards these tokens.
mandoc discards these tokens.
```

```
249 The
250 .Cm back Ar n ,
251 .Cm fwd Ar n ,
252 .Cm up Ar n ,
253 and
254 .Cm down Ar n
255 commands are also ignored.
256 .El
257 .Sh SEE ALSO
258 .Xr mandoc 1 ,
259 .Xr man 5 ,
243 .Xr man 7 ,
260 .Xr mandoc_char 5 ,
261 .Xr mdoc 5 ,
262 .Xr roff 5
263 .Rs
264 .%A Brian W. Kernighan
265 .%A Lorinda L. Cherry
266 .%T System for Typesetting Mathematics
267 .%J Communications of the ACM
268 .%V 18
269 .%P 151\(en157
270 .%D March, 1975
271 .Re
272 .Rs
273 .%A Brian W. Kernighan
274 .%A Lorinda L. Cherry
275 .%T Typesetting Mathematics, User's Guide
276 .%D 1976
277 .Re
278 .Rs
279 .%A Brian W. Kernighan
280 .%A Lorinda L. Cherry
281 .%T Typesetting Mathematics, User's Guide (Second Edition)
282 .%D 1978
283 .Re
284 .Sh HISTORY
285 The eqn utility, a preprocessor for troff, was originally written by
286 Brian W. Kernighan and Lorinda L. Cherry in 1975.
287 The GNU reimplementation of eqn, part of the GNU troff package, was
288 released in 1989 by James Clark.
289 The eqn component of
290 .Xr mandoc 1
291 was added in 2011.
292 .Sh AUTHORS
293 This
294 .Nm
295 reference was written by
296 .An Kristaps Dzonsons ,
297 .Mt kristaps@bsd.lv .
```

```
**********************************************************
    8697 Sat Jul 19 14:19:58 2014
new/usr/src/man/man5/man.5
Latest round of fixes per RM and AL.  Fix bugs found in man.c.
**********************************************************
```

```
.\" Copyright (c) 1995, Sun Microsystems, Inc.
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.Dd "Jul 19, 2014"
.\"
.\" Permission to use, copy, modify, and distribute this software for any
.\" purpose with or without fee is hereby granted, provided that the above
.\" copyright notice and this permission notice appear in all copies.
.\"
.\" THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
.\" WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
.\" MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
.\" ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
.\" WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
.\" ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
.\" OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
.\"
.\"
.\" Copyright (c) 2009, 2010, 2011 Kristaps Dzonsons <kristaps@bsd.lv>
.\" Copyright (c) 2011 Ingo Schwarze <schwarze@openbsd.org>
.\" Copyright 2012 Nexenta Systems, Inc. All rights reserved.
.\"
.Dd Jan 3, 2012
.Dt MAN 5
.Os
.Sh NAME
.Nm man
.Nd macros to format Reference Manual pages
.Sh SYNOPSIS
.Nm mandoc
.Fl T Ar man
.Ar
.Nm nroff
.Fl man
.Ar
.Nm troff
.Fl man
.Ar
.Nd legacy formatting language for manual pages
.Sh DESCRIPTION
These macros are used to lay out the reference pages in this manual. Note: if
.Ar file
contains format input for a preprocessor, the commands shown
above must be piped through the appropriate preprocessor. This is handled
automatically by the
Traditionally, the
.Nm man
language has been used to write
.Ux
manuals for the
.Xr man 1
command. See the
.Sx Conventions
section.
.Lp
Any text argument
.Ar t
may be zero to six words. Quotes may be used to
include SPACE characters in a
.Qq word .
```



```
If
.Ar text
is empty, the special
treatment is applied to the next input line with text to be printed. In this
way
.Nm \&.I
may be used to italicize a whole line, or
.Nm \&.SB
may be used to make small bold letters.
.Lp
A prevailing indent distance is remembered between successive indented
paragraphs, and is reset to default value upon reaching a non-indented
paragraph.  Default units for indents
.Nm i
are ens.
.Lp
Type font and size are reset to default values before each paragraph, and after
processing font and size setting macros.
utility.
It supports limited control of presentational details like fonts,
indentation and spacing.
This reference document describes the structure of manual pages
and the syntax and usage of the man language.
.Pp
These strings are predefined by
.Nm -man :
.Bl -tag -width Ds
.It Nm \e*R
.Sq \(rg ,
.Sq (Reg)
in
.Nm nroff .
.It Nm \e*S
Change to default type size.
.Bf -emphasis
Do not use
.Nm
to write your manuals:
.Ef
It lacks support for semantic markup.
Use the
.Xr mdoc 5
language, instead.
.Pp
In a
.Nm
document, lines beginning with the control character
.Sq \&.
are called
.Dq macro lines .
The first word is the macro name.
It usually consists of two capital letters.
For a list of available macros, see
.Sx MACRO OVERVIEW .
The words following the macro name are arguments to the macro.
.Pp
Lines not beginning with the control character are called
.Dq text lines .
They provide free-form text to be printed; the formatting of the text
depends on the respective processing context:
.Bd -literal -offset indent
\&.SH Macro lines change control state.
Text lines are interpreted within the current state.
.Ed
.Pp
Many aspects of the basic syntax of the
```

```
 70 .Nm
 71 language are based on the
 72 .Xr roff 5
 73 language; see the
 74 .Em LANGUAGE SYNTAX
 75 and
 76 .Em MACRO SYNTAX
 77 sections in the
 78 .Xr roff 5
 79 manual for details, in particular regarding
 80 comments, escape sequences, whitespace, and quoting.
 81 .Sh MANUAL STRUCTURE
 82 Each
 83 .Nm
 84 document must contain the
 85 .Sx \&TH
 86 macro describing the document's section and title.
 87 It may occur anywhere in the document, although conventionally it
 88 appears as the first macro.
 89 .Pp
 90 Beyond
 91 .Sx \&TH ,
 92 at least one macro or text line must appear in the document.
 93 .Pp
 94 The following is a well-formed skeleton
 95 .Nm
 96 file for a utility
 97 .Qq progname :
 98 .Bd -literal -offset indent
 99 \&.TH PROGNAME 1 "Oct 10, 2009"
100 \&.SH NAME
101 \efBprogname\efR \e(en a description goes here
102 \&.\e\(dq .SH LIBRARY
103 \&.\e\(dq For sections 2 & 3 only.
104 \&.SH SYNOPSIS
105 \efBprogname\efR [\efB\e-options\efR] arguments...
106 \&.SH DESCRIPTION
107 The \efBfoo\efR utility processes files...
108 \&.\e\(dq .SH IMPLEMENTATION NOTES
109 \&.\e\(dq .SH RETURN VALUES
110 \&.\e\(dq For sections 2, 3, & 9 only.
111 \&.\e\(dq .SH ENVIRONMENT
112 \&.\e\(dq For sections 1, 1M, 5, & 6 only.
113 \&.\e\(dq .SH FILES
114 \&.\e\(dq .SH EXIT STATUS
115 \&.\e\(dq For sections 1, 1M, & 6 only.
116 \&.\e\(dq .SH EXAMPLES
117 \&.\e\(dq .SH DIAGNOSTICS
118 \&.\e\(dq For sections 1, 1M, 5, 6, & 7 only.
119 \&.\e\(dq .SH ERRORS
120 \&.\e\(dq For sections 2, 3, & 9 only.
121 \&.\e\(dq .SH SEE ALSO
122 \&.\e\(dq .BR foo ( 1 )
123 \&.\e\(dq .SH STANDARDS
124 \&.\e\(dq .SH HISTORY
125 \&.\e\(dq .SH AUTHORS
126 \&.\e\(dq .SH CAVEATS
127 \&.\e\(dq .SH BUGS
128 \&.\e\(dq .SH SECURITY CONSIDERATIONS
129 .Ed
130 .Pp
131 The sections in a
132 .Nm
133 document are conventionally ordered as they appear above.
134 Sections should be composed as follows:
135 .Bl -ohang -offset indent
```



```
136 .It Em NAME
137 The name(s) and a short description of the documented material.
138 The syntax for this is generally as follows:
139 .Pp
140 .D1 \efBname\efR \e(en description
141 .It Em LIBRARY
142 The name of the library containing the documented material, which is
143 assumed to be a function in a section 2 or 3 manual.
144 For functions in the C library, this may be as follows:
145 .Pp
146 .D1 Standard C Library (libc, -lc)
147 .It Em SYNOPSIS
148 Documents the utility invocation syntax, function call syntax, or device
149 configuration.
150 .Pp
151 For the first, utilities (sections 1, 1M, and 6), this is
152 generally structured as follows:
153 .Pp
154 .D1 \efBname\efR [-\efBab\efR] [-\efBc\efR\efIarg\efR] \efBpath\efR...
155 .Pp
156 For the second, function calls (sections 2, 3, 9):
157 .Pp
158 .D1 \&.B char *name(char *\efIarg\efR);
159 .Pp
160 And for the third, configurations (section 7):
161 .Pp
162 .D1 \&.B name* at cardbus ? function ?
163 .Pp
164 Manuals not in these sections generally don't need a
165 .Em SYNOPSIS .
166 .It Em DESCRIPTION
167 This expands upon the brief, one-line description in
168 .Em NAME .
169 It usually contains a break-down of the options (if documenting a
170 command).
171 .It Em IMPLEMENTATION NOTES
172 Implementation-specific notes should be kept here.
173 This is useful when implementing standard functions that may have side
174 effects or notable algorithmic implications.
175 .It Em RETURN VALUES
176 This section documents the return values of functions in sections 2, 3, and 9.
177 .It Em ENVIRONMENT
178 Documents any usages of environment variables, e.g.,
179 .Xr environ 5 .
180 .It Em FILES
181 Documents files used.
182 It's helpful to document both the file name and a short description of how
183 the file is used (created, modified, etc.).
184 .It Em EXIT STATUS
185 This section documents the command exit status for
186 section 1, 6, and 8 utilities.
187 Historically, this information was described in
188 .Em DIAGNOSTICS ,
189 a practise that is now discouraged.
190 .It Em EXAMPLES
191 Example usages.
192 This often contains snippets of well-formed,
193 well-tested invocations.
194 Make sure that examples work properly!
195 .It Em DIAGNOSTICS
196 Documents error conditions.
197 This is most useful in section 4 manuals.
198 Historically, this section was used in place of
199 .Em EXIT STATUS
200 for manuals in sections 1, 6, and 8; however, this practise is
201 discouraged.
```

```
.It Em ERRORS
Documents error handling in sections 2, 3, and 9.
.It Em SEE ALSO
References other manuals with related topics.
This section should exist for most manuals.
.Pp
.D1 \&.BR bar \&( 1 \&),
.Pp
Cross-references should conventionally be ordered
first by section, then alphabetically.
.It Em STANDARDS
References any standards implemented or used, such as
.Pp
.D1 IEEE Std 1003.2 (\e(lqPOSIX.2\e(rq)
.Pp
If not adhering to any standards, the
.Em HISTORY
section should be used.
.It Em HISTORY
A brief history of the subject, including where support first appeared.
.It Em AUTHORS
Credits to the person or persons who wrote the code and/or documentation.
Authors should generally be noted by both name and email address.
.It Em CAVEATS
Common misuses and misunderstandings should be explained
in this section.
.It Em BUGS
Known bugs, limitations, and work-arounds should be described
in this section.
.It Em SECURITY CONSIDERATIONS
Documents any security precautions that operators should consider.
.El
.Sh "Requests"
* n.t.l. = next text line; p.i. = prevailing indent
.Bl -column ".TH n s d f m" "Cause " "t=n.t.l.*" "Explanation " -offset Ds
.It Sy Request  Sy Cause        Sy "If No"      Sy Explanation
.It ""  Sy Break        Sy "Argument"   ""
.It Nm \&.B Ar "t"      no      Ar t Ns =n.t.l.*        Text is in bold font.
.It Nm \&.BI Ar t       no      Ar t Ns =n.t.l. Join words, alternating bold and
.It Nm \&.BR Ar t       no      Ar t Ns =n.t.l. Join words, alternating bold and
.It Nm \&.DT    no      Li \&.5i 1i...  Restore default tabs.
.It Nm \&.HP Ar i       yes     Ar i Ns =p.i.*  "Begin paragraph with hanging in
.It Nm \&.I Ar t        no      Ar t Ns =n.t.l. Text is italic.
.It Nm \&.IB Ar t       no      Ar t Ns =n.t.l. Join words, altenrating italic a
.It Nm \&.IP Ar x Ar i  yes     Ar x Ns =""     Same as
.Nm \&.TP
with tag
.Ar x .
.It Nm \&.IR Ar t       no      Ar t Ns =n.t.l. Join words, alternating italic a
.It Nm \&.IX Ar t       no      -       Index macro, not used (obsolete).
.It Nm \&.LP    yes     -       Begin left-aligned paragraph. Set prevailing ind
.It Nm \&.P     yes     -       Same as
.Nm \&.LP .
.It Nm \&.PD Ar d       no      Ar d Ns =.4v    Set vertical distance between pa
.It Nm \&.PP    yes     -       Same as
.Nm \&.LP .
.It Nm \&.RE    yes     -       End of relative indent. Restores prevailing inde
.It Nm \&.RB Ar t       no      Ar t Ns =n.t.l. Join words, alternating roman an
.It Nm \&.RI Ar t       no      Ar t Ns =n.t.l. Join words, alternating roman an
.It Nm \&.RS Ar i       yes     Ar i Ns =p.i.   Start relative indent, increase
Sets prevailing indent to .5i for nested indents.
.It Nm \&.SB Ar t       no      -       Reduce size of text by 1 point, make tex
.It Nm \&.SH Ar t       yes     -       Section Heading.
.It Nm \&.SM Ar t       no      Ar t Ns =n.t.l. Reduce size of text by 1 point.
.It Nm \&.SS Ar t       yes     Ar t Ns =n.t.l. Section Subheading.
.It Nm \&.TH Ar n s d f m       yes     -       Begin reference page Ar n , No o
```



```
.It Nm \&.TP Ar i       yes     Ar i Ns =p.i.   Begin indented paragraph, with t
.Ar i .
.It Nm \&.TX Ar t p     no      -       Resolve the title abbreviation Ar t ; No
.Sh MACRO OVERVIEW
This overview is sorted such that macros of similar purpose are listed
together, to help find the best macro for any given purpose.
Deprecated macros are not included in the overview, but can be found
in the alphabetical reference below.
.Ss Page header and footer meta-data
.Bl -column "PP, LP, P" description
.It Sx TH Ta set the title: Ar title section date Op Ar source Op Ar volume
.It Sx AT Ta display AT&T UNIX version in the page footer (<= 1 argument)
.It Sx UC Ta display BSD version in the page footer (<= 1 argument)
.El
.Ss "Conventions"
When formatting a manual page,
.Nm
examines the first line to determine
whether it requires special processing. For example a first line consisting of:
.Lp
.Dl \&'\e" t
.Lp
indicates that the manual page must be run through the
.Xr tbl 1
preprocessor.
.Lp
A typical manual page for a command or function is laid out as follows:
.Bl -tag -width ".SH RETURN VALUES"
.
.It Nm \&.TH Ar title Op "1-9"
.
The name of the command or function, which serves as the title of the manual
page. This is followed by the number of the section in which it appears.
.
.It Nm SH NAME
.
The name, or list of names, by which the command is called, followed by a dash
and then a one-line summary of the action performed. All in roman font, this
section contains no
.Xr troff 1
commands or escapes, and no macro requests.
It is used to generate the database used by the
.Xr whatis 1
command.
.
.It Nm SH SYNOPSIS
.Bl -tag -width "Functions:"
.It Sy Commands:
The syntax of the command and its arguments, as typed on the command line.
When in boldface, a word must be typed exactly as printed.  When in italics, a
word can be replaced with an argument that you supply. References to bold or
italicized items are not capitalized in other sections, even when they begin a
sentence.
.Lp
Syntactic symbols appear in roman face:
.Bl -tag -width "   "
.It Op " "
An argument, when surrounded by brackets is optional.
.It |
Arguments separated by a vertical bar are exclusive. You can supply only one
item from such a list.
.It \&.\|.\|.
Arguments followed by an ellipsis can be repeated. When an ellipsis follows a
bracketed set, the expression within the brackets can be repeated.
.Ss Sections and paragraphs
.Bl -column "PP, LP, P" description
```

```
.It Sx SH Ta section header (one line)
.It Sx SS Ta subsection header (one line)
.It Sx PP , LP , P Ta start an undecorated paragraph (no arguments)
.It Sx RS , RE Ta reset the left margin: Op Ar width
.It Sx IP Ta indented paragraph: Op Ar head Op Ar width
.It Sx TP Ta tagged paragraph: Op Ar width
.It Sx HP Ta hanged paragraph: Op Ar width
.It Sx \&br Ta force output line break in text mode (no arguments)
.It Sx \&sp Ta force vertical space: Op Ar height
.It Sx fi , nf Ta fill mode and no-fill mode (no arguments)
.It Sx in Ta additional indent: Op Ar width
.El
.It Sy Functions:
If required, the data declaration, or
.Li #include
directive, is shown first,
followed by the  function declaration. Otherwise, the function declaration is
shown.
.Ss Physical markup
.Bl -column "PP, LP, P" description
.It Sx B Ta boldface font
.It Sx I Ta italic font
.It Sx R Ta roman (default) font
.It Sx SB Ta small boldface font
.It Sx SM Ta small roman font
.It Sx BI Ta alternate between boldface and italic fonts
.It Sx BR Ta alternate between boldface and roman fonts
.It Sx IB Ta alternate between italic and boldface fonts
.It Sx IR Ta alternate between italic and roman fonts
.It Sx RB Ta alternate between roman and boldface fonts
.It Sx RI Ta alternate between roman and italic fonts
.El
.
.It Nm \&.SH DESCRIPTION
.
A narrative overview of the command or function’s external behavior. This
includes how it interacts with files or data, and how it handles the standard
input, standard output and standard error. Internals and implementation details
are normally omitted. This section attempts to provide a succinct overview in
answer to the question, "what does it do?"
.Lp
Literal text from the synopsis appears in constant width, as do literal
filenames and references to items that appear elsewhere in the  reference
manuals. Arguments are italicized.
.Lp
If a command interprets either subcommands or an input grammar, its command
interface or input grammar is normally described in a
.Nm USAGE
section, which follows the
.Nm OPTIONS
section.  The
.Nm DESCRIPTION
section only
describes the behavior of the command itself, not that of subcommands.
.
.It Nm \&.SH OPTIONS
.
The list of options along with a description of how each affects the command’s
operation.
.
.It Nm \&.SH RETURN VALUES
.
A list of the values the library routine will return to the calling  program
and the conditions that cause these values to be returned.
.
.It Nm \&.SH EXIT STATUS
```



```
.
A list of the values the utility will return to the calling  program or shell,
and the conditions that cause these values to be  returned.
.
.It Nm \&.SH FILES
.
A list of files associated with the command or function.
.
.It Nm \&.SH SEE ALSO
.
A comma-separated list of related manual pages, followed by references to other
published materials.
.
.It Nm \&.SH DIAGNOSTICS
.
A list of diagnostic messages and an explanation of each.
.
.It Nm \&.SH BUGS
.
A description of limitations, known defects, and possible problems associated
with the command or function.
.Ss Semantic markup
.Bl -column "PP, LP, P" description
.It Sx OP Ta optional arguments
.El
.Sh FILES
.Pa /usr/share/man/whatis
.Sh NOTES
.Sh MACRO REFERENCE
This section is a canonical reference to all macros, arranged
alphabetically.
For the scoping of individual macros, see
.Sx MACRO SYNTAX .
.Ss \&AT
Sets the volume for the footer for compatibility with man pages from
.Tn AT&T UNIX
releases.
The optional arguments specify which release it is from.
.Ss \&B
Text is rendered in bold face.
.Pp
See also
.Sx \&I
and
.Sx \&R .
.Ss \&BI
Text is rendered alternately in bold face and italic.
Thus,
.Sq .BI this word and that
causes
.Sq this
and
.Sq and
to render in bold face, while
.Sq word
and
.Sq that
render in italics.
Whitespace between arguments is omitted in output.
.Pp
Examples:
.Pp
.Dl \&.BI bold italic bold italic
.Pp
The output of this example will be emboldened
.Dq bold
```

```
315 and italicised
316 .Dq italic ,
317 with spaces stripped between arguments.
318 .Pp
319 See also
320 .Sx \&IB ,
321 .Sx \&BR ,
322 .Sx \&RB ,
323 .Sx \&RI ,
324 and
325 .Sx \&IR .
326 .Ss \&BR
327 Text is rendered alternately in bold face and roman (the default font).
328 Whitespace between arguments is omitted in output.
329 .Pp
330 See
331 .Sx \&BI
332 for an equivalent example.
333 .Pp
334 See also
335 .Sx \&BI ,
336 .Sx \&IB ,
337 .Sx \&RB ,
338 .Sx \&RI ,
339 and
340 .Sx \&IR .
341 .Ss \&DT
342 Has no effect.
343 Included for compatibility.
344 .Ss \&HP
345 Begin a paragraph whose initial output line is left-justified, but
346 subsequent output lines are indented, with the following syntax:
347 .Bd -filled -offset indent
348 .Pf \. Sx \&HP
349 .Op Cm width
350 .Ed
351 .Pp
222 The
353 .Cm width
354 argument must conform to
355 .Sx Scaling Widths .
356 If specified, it’s saved for later paragraph left-margins; if unspecified, the
357 saved or default width is used.
358 .Pp
359 See also
360 .Sx \&IP ,
361 .Sx \&LP ,
362 .Sx \&P ,
363 .Sx \&PP ,
364 and
365 .Sx \&TP .
366 .Ss \&I
367 Text is rendered in italics.
368 .Pp
369 See also
370 .Sx \&B
371 and
372 .Sx \&R .
373 .Ss \&IB
374 Text is rendered alternately in italics and bold face.
375 Whitespace between arguments is omitted in output.
376 .Pp
377 See
378 .Sx \&BI
379 for an equivalent example.
380 .Pp
```



```
381 See also
382 .Sx \&BI ,
383 .Sx \&BR ,
384 .Sx \&RB ,
385 .Sx \&RI ,
386 and
387 .Sx \&IR .
388 .Ss \&IP
389 Begin an indented paragraph with the following syntax:
390 .Bd -filled -offset indent
391 .Pf \. Sx \&IP
392 .Op Cm head Op Cm width
393 .Ed
394 .Pp
395 The
396 .Cm width
397 argument defines the width of the left margin and is defined by
398 .Sx Scaling Widths .
399 It’s saved for later paragraph left-margins; if unspecified, the saved or
400 default width is used.
401 .Pp
402 The
403 .Cm head
404 argument is used as a leading term, flushed to the left margin.
405 This is useful for bulleted paragraphs and so on.
406 .Pp
407 See also
408 .Sx \&HP ,
409 .Sx \&LP ,
410 .Sx \&P ,
411 .Sx \&PP ,
412 and
413 .Sx \&TP .
414 .Ss \&IR
415 Text is rendered alternately in italics and roman (the default font).
416 Whitespace between arguments is omitted in output.
417 .Pp
418 See
419 .Sx \&BI
420 for an equivalent example.
421 .Pp
422 See also
423 .Sx \&BI ,
424 .Sx \&IB ,
425 .Sx \&BR ,
426 .Sx \&RB ,
427 and
428 .Sx \&RI .
429 .Ss \&LP
430 Begin an undecorated paragraph.
431 The scope of a paragraph is closed by a subsequent paragraph,
432 sub-section, section, or end of file.
433 The saved paragraph left-margin width is reset to the default.
434 .Pp
435 See also
436 .Sx \&HP ,
437 .Sx \&IP ,
438 .Sx \&P ,
439 .Sx \&PP ,
440 and
441 .Sx \&TP .
442 .Ss \&OP
443 Optional command-line argument.
444 This has the following syntax:
445 .Bd -filled -offset indent
446 .Pf \. Sx \&OP
```

```
.Cm key Op Cm value
.Ed
.Pp
The
.Cm key
is usually a command-line flag and
.Cm value
its argument.
.Ss \&P
Synonym for
.Sx \&LP .
.Pp
See also
.Sx \&HP ,
.Sx \&IP ,
.Sx \&LP ,
.Sx \&PP ,
and
.Sx \&TP .
.Ss \&PP
Synonym for
.Sx \&LP .
.Pp
See also
.Sx \&HP ,
.Sx \&IP ,
.Sx \&LP ,
.Sx \&P ,
and
.Sx \&TP .
.Ss \&R
Text is rendered in roman (the default font).
.Pp
See also
.Sx \&I
and
.Sx \&B .
.Ss \&RB
Text is rendered alternately in roman (the default font) and bold face.
Whitespace between arguments is omitted in output.
.Pp
See
.Sx \&BI
for an equivalent example.
.Pp
See also
.Sx \&BI ,
.Sx \&IB ,
.Sx \&BR ,
.Sx \&RI ,
and
.Sx \&IR .
.Ss \&RE
Explicitly close out the scope of a prior
.Sx \&RS .
The default left margin is restored to the state of the original
.Sx \&RS
invocation.
.Ss \&RI
Text is rendered alternately in roman (the default font) and italics.
Whitespace between arguments is omitted in output.
.Pp
See
.Sx \&BI
for an equivalent example.
.Pp
```



```
See also
.Sx \&BI ,
.Sx \&IB ,
.Sx \&BR ,
.Sx \&RB ,
and
.Sx \&IR .
.Ss \&RS
Temporarily reset the default left margin.
This has the following syntax:
.Bd -filled -offset indent
.Pf \. Sx \&RS
.Op Cm width
.Ed
.Pp
The
.Cm width
argument must conform to
.Sx Scaling Widths .
If not specified, the saved or default width is used.
.Pp
See also
.Sx \&RE .
.Ss \&SB
Text is rendered in small size (one point smaller than the default font)
bold face.
.Ss \&SH
Begin a section.
The scope of a section is only closed by another section or the end of
file.
The paragraph left-margin width is reset to the default.
.Ss \&SM
Text is rendered in small size (one point smaller than the default
font).
.Ss \&SS
Begin a sub-section.
The scope of a sub-section is closed by a subsequent sub-section,
section, or end of file.
The paragraph left-margin width is reset to the default.
.Ss \&TH
Sets the title of the manual page with the following syntax:
.Bd -filled -offset indent
.Pf \. Sx \&TH
.Ar title section date
.Op Ar source Op Ar volume
.Ed
.Pp
Conventionally, the document
.Ar title
is given in all caps.
The recommended
.Ar date
format is
.Sy YYYY-MM-DD
as specified in the ISO-8601 standard;
if the argument does not conform, it is printed verbatim.
If the
.Ar date
is empty or not specified, the current date is used.
The optional
.Ar source
string specifies the organisation providing the utility.
The
.Ar volume
string replaces the default rendered volume, which is dictated by the
manual section.
```

```
.Pp
Examples:
.Pp
.Dl \&.TH CVS 5 "1992-02-12" GNU
.Ss \&TP
Begin a paragraph where the head, if exceeding the indentation width, is
followed by a newline; if not, the body follows on the same line after a
buffer to the indentation width.
Subsequent output lines are indented.
The syntax is as follows:
.Bd -filled -offset indent
.Pf \. Sx \&TP
.Op Cm width
.Ed
.Pp
The
.Cm width
argument must conform to
.Sx Scaling Widths .
If specified, it’s saved for later paragraph left-margins; if
unspecified, the saved or default width is used.
.Pp
See also
.Sx \&HP ,
.Sx \&IP ,
.Sx \&LP ,
.Sx \&P ,
and
.Sx \&PP .
.Ss \&UC
Sets the volume for the footer for compatibility with man pages from
BSD releases.
The optional first argument specifies which release it is from.
.Ss \&br
Breaks the current line.
Consecutive invocations have no further effect.
.Pp
See also
.Sx \&sp .
.Ss \&fi
End literal mode begun by
.Sx \&nf .
.Ss \&ft
Change the current font mode.
See
.Sx Text Decoration
for a listing of available font modes.
.Ss \&in
Indent relative to the current indentation:
.Pp
.D1 Pf \. Sx \&in Op Cm width
.Pp
If
.Cm width
is signed, the new offset is relative.
Otherwise, it is absolute.
This value is reset upon the next paragraph, section, or sub-section.
.Ss \&na
Don’t align to the right margin.
.Ss \&nf
Begin literal mode: all subsequent free-form lines have their end of
line boundaries preserved.
May be ended by
.Sx \&fi .
Literal mode is implicitly ended by
.Sx \&SH
```



```
or
.Sx \&SS .
.Ss \&sp
Insert vertical spaces into output with the following syntax:
.Bd -filled -offset indent
.Pf \. Sx \&sp
.Op Cm height
.Ed
.Pp
Insert
.Cm height
spaces, which must conform to
.Sx Scaling Widths .
If 0, this is equivalent to the
.Sx \&br
macro.
Defaults to 1, if unspecified.
.Pp
See also
.Sx \&br .
.Sh MACRO SYNTAX
The
.Nm
package should not be used for new documentation.  The
.Xr mdoc 5 ,
package is preferred, as it uses semantic markup rather than physical markup.
.Sh CODE SET INDEPENDENCE
When processed with
.Xr mandoc 1 ,
this package is Code Set Independent. However, when processed with
legacy tools such as
.Xr nroff 1
macros are classified by scope: line scope or block scope.
Line macros are only scoped to the current line (and, in some
situations, the subsequent line).
Block macros are scoped to the current line and subsequent lines until
closed by another block macro.
.Ss Line Macros
Line macros are generally scoped to the current line, with the body
consisting of zero or more arguments.
If a macro is scoped to the next line and the line arguments are empty,
the next line, which must be text, is used instead.
Thus:
.Bd -literal -offset indent
\&.I
foo
.Ed
.Pp
is equivalent to
.Sq \&.I foo .
If next-line macros are invoked consecutively, only the last is used.
If a next-line macro is followed by a non-next-line macro, an error is
raised, except for
.Sx \&br ,
.Sx \&sp ,
and
.Xr troff 1 ,
the use of multi-byte characters may not be supported.
.Sh INTERFACE STABILITY
.Nm Obsolete Committed .
.Sx \&na .
.Pp
The syntax is as follows:
.Bd -literal -offset indent
\&.YO \(lBbody...\(rB
\(lBbody...\(rB
```

```
.Ed
.Bl -column "MacroX" "ArgumentsX" "ScopeXXXXX" "CompatX" -offset indent
.It Em Macro Ta Em Arguments Ta Em Scope     Ta Em Notes
.It Sx \&AT  Ta    <=1       Ta    current   Ta    \&
.It Sx \&B   Ta    n         Ta    next-line Ta    \&
.It Sx \&BI  Ta    n         Ta    current   Ta    \&
.It Sx \&BR  Ta    n         Ta    current   Ta    \&
.It Sx \&DT  Ta    0         Ta    current   Ta    \&
.It Sx \&I   Ta    n         Ta    next-line Ta    \&
.It Sx \&IB  Ta    n         Ta    current   Ta    \&
.It Sx \&IR  Ta    n         Ta    current   Ta    \&
.It Sx \&OP  Ta    0, 1      Ta    current   Ta    compat
.It Sx \&R   Ta    n         Ta    next-line Ta    \&
.It Sx \&RB  Ta    n         Ta    current   Ta    \&
.It Sx \&RI  Ta    n         Ta    current   Ta    \&
.It Sx \&SB  Ta    n         Ta    next-line Ta    \&
.It Sx \&SM  Ta    n         Ta    next-line Ta    \&
.It Sx \&TH  Ta    >1, <6    Ta    current   Ta    \&
.It Sx \&UC  Ta    <=1       Ta    current   Ta    \&
.It Sx \&br  Ta    0         Ta    current   Ta    compat
.It Sx \&fi  Ta    0         Ta    current   Ta    compat
.It Sx \&ft  Ta    1         Ta    current   Ta    compat
.It Sx \&in  Ta    1         Ta    current   Ta    compat
.It Sx \&na  Ta    0         Ta    current   Ta    compat
.It Sx \&nf  Ta    0         Ta    current   Ta    compat
.It Sx \&sp  Ta    1         Ta    current   Ta    compat
.El
.Pp
Macros marked as
.Qq compat
are included for compatibility with the significant corpus of existing
manuals that mix dialects of roff.
These macros should not be used for portable
.Nm
manuals.
.Ss Block Macros
Block macros comprise a head and body.
As with in-line macros, the head is scoped to the current line and, in
one circumstance, the next line (the next-line stipulations as in
.Sx Line Macros
apply here as well).
.Pp
The syntax is as follows:
.Bd -literal -offset indent
\&.YO \(lBhead...\(rB
\(lBhead...\(rB
\(lBbody...\(rB
.Ed
.Pp
The closure of body scope may be to the section, where a macro is closed
by
.Sx \&SH ;
sub-section, closed by a section or
.Sx \&SS ;
part, closed by a section, sub-section, or
.Sx \&RE ;
or paragraph, closed by a section, sub-section, part,
.Sx \&HP ,
.Sx \&IP ,
.Sx \&LP ,
.Sx \&P ,
.Sx \&PP ,
or
.Sx \&TP .
No closure refers to an explicit block closing macro.
.Pp
```



```
As a rule, block macros may not be nested; thus, calling a block macro
while another block macro scope is open, and the open scope is not
implicitly closed, is syntactically incorrect.
.Bl -column "MacroX" "ArgumentsX" "Head ScopeX" "sub-sectionX" "compatX" -offset
.It Em Macro Ta Em Arguments Ta Em Head Scope Ta Em Body Scope  Ta Em Notes
.It Sx \&HP  Ta    <2        Ta    current    Ta    paragraph   Ta    \&
.It Sx \&IP  Ta    <3        Ta    current    Ta    paragraph   Ta    \&
.It Sx \&LP  Ta    0         Ta    current    Ta    paragraph   Ta    \&
.It Sx \&P   Ta    0         Ta    current    Ta    paragraph   Ta    \&
.It Sx \&PP  Ta    0         Ta    current    Ta    paragraph   Ta    \&
.It Sx \&RE  Ta    0         Ta    current    Ta    none        Ta    compat
.It Sx \&RS  Ta    1         Ta    current    Ta    part        Ta    compat
.It Sx \&SH  Ta    >0        Ta    next-line  Ta    section     Ta    \&
.It Sx \&SS  Ta    >0        Ta    next-line  Ta    sub-section Ta    \&
.It Sx \&TP  Ta    n         Ta    next-line  Ta    paragraph   Ta    \&
.El
.Pp
Macros marked
.Qq compat
are as mentioned in
.Sx Line Macros .
.Pp
If a block macro is next-line scoped, it may only be followed by in-line
macros for decorating text.
.Ss Font handling
In
.Nm
documents, both
.Sx Physical markup
macros and
.Xr roff 5
.Ql \ef
font escape sequences can be used to choose fonts.
In text lines, the effect of manual font selection by escape sequences
only lasts until the next macro invocation; in macro lines, it only lasts
until the end of the macro scope.
Note that macros like
.Sx \&BR
open and close a font scope for each argument.
.Sh COMPATIBILITY
This section documents areas of questionable portability between
implementations of the
.Nm
language.
.Pp
.Bl -dash -compact
.It
Do not depend on
.Sx \&SH
or
.Sx \&SS
to close out a literal context opened with
.Sx \&nf .
This behaviour may not be portable.
.It
In quoted literals, GNU troff allowed pair-wise double-quotes to produce
a standalone double-quote in formatted output.
It is not known whether this behaviour is exhibited by other formatters.
.It
troff suppresses a newline before
.Sq \(aq
macro output; in mandoc, it is an alias for the standard
.Sq \&.
control character.
.It
The
```

```
239 .Xr mdoc 5
240 package should be used instead.
830 .Sq \eh
831 .Pq horizontal position ,
832 .Sq \ev
833 .Pq vertical position ,
834 .Sq \em
835 .Pq text colour ,
836 .Sq \eM
837 .Pq text filling colour ,
838 .Sq \ez
839 .Pq zero-length character ,
840 .Sq \ew
841 .Pq string length ,
842 .Sq \ek
843 .Pq horizontal position marker ,
844 .Sq \eo
845 .Pq text overstrike ,
846 and
847 .Sq \es
848 .Pq text size
849 escape sequences are all discarded in mandoc.
850 .It
851 The
852 .Sq \ef
853 scaling unit is accepted by mandoc, but rendered as the default unit.
854 .It
855 The
856 .Sx \&sp
857 macro does not accept negative values in mandoc.
858 In GNU troff, this would result in strange behaviour.
859 .It
860 In page header lines, GNU troff versions up to and including 1.21
861 only print
862 .Ar volume
863 names explicitly specified in the
864 .Sx \&TH
865 macro; mandoc and newer groff print the default volume name
866 corresponding to the
867 .Ar section
868 number when no
869 .Ar volume
870 is given, like in
871 .Xr mdoc 5 .
872 .El
873 .Pp
874 The
875 .Sx OP
876 macro is part of the extended
877 .Nm
878 macro set, and may not be portable to non-GNU troff implementations.
879 .Sh INTERFACE STABILITY
880 .Nm "Obsolete Committed" .
241 .Sh SEE ALSO
242 .Xr eqn 1 ,
243 .Xr man 1 ,
244 .Xr mandoc 1 ,
245 .Xr nroff 1 ,
246 .Xr troff 1 ,
247 .Xr tbl 1 ,
248 .Xr whatis 1 ,
884 .Xr eqn 5 ,
885 .Xr mandoc_char 5 ,
249 .Xr mdoc 5 ,
250 .Rs
251 .%A Dale Dougherty and Tim O’Reilly
```



```
252 .%B Unix Text Processing
253 .Re
887 .Xr roff 5 ,
888 .Xr tbl 5
889 .Sh HISTORY
890 The
891 .Nm
892 language first appeared as a macro package for the roff typesetting
893 system in
894 .At v7 .
895 It was later rewritten by James Clark as a macro package for groff.
896 Eric S. Raymond wrote the extended
897 .Nm
898 macros for groff in 2007.
899 The stand-alone implementation that is part of the
900 .Xr mandoc 1
901 utility written by Kristaps Dzonsons appeared in
902 .Ox 4.6 .
903 .Sh AUTHORS
904 This
905 .Nm
906 reference was written by
907 .An Kristaps Dzonsons ,
908 .Mt kristaps@bsd.lv .
909 .Sh CAVEATS
910 Do not use this language.
911 Use
912 .Xr mdoc 5 ,
913 instead.
```

```
***********************************************************
   75905 Sat Jul 19 14:19:59 2014
new/usr/src/man/man5/mdoc.5
Latest round of fixes per RM and AL.  Fix bugs found in man.c.
***********************************************************
```

```
.\"
.\" Permission to use, copy, modify, and distribute this software for any
.\" purpose with or without fee is hereby granted, provided that the above
.\" copyright notice and this permission notice appear in all copies.
.\"
.\" THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
.\" WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
.\" MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
.\" ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
.\" WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
.\" ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
.\" OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
.\"
.\"
.\" Copyright (c) 2009, 2010, 2011 Kristaps Dzonsons <kristaps@bsd.lv>
.\" Copyright (c) 2010, 2011 Ingo Schwarze <schwarze@openbsd.org>
.\" Copyright 2012 Nexenta Systems, Inc. All rights reserved.
.\" Copyright 2014 Garrett D'Amore <garrett@dmaore.org>
.\"
.Dd Jul 19, 2014
.Dd Jul 16, 2014
.Dt MDOC 5
.Os
.Sh NAME
.Nm mdoc
.Nd semantic markup language for formatting manual pages
.Sh DESCRIPTION
The
.Nm mdoc
language supports authoring of manual pages for the
.Xr man 1
utility by allowing semantic annotations of words, phrases,
page sections and complete manual pages.
Such annotations are used by formatting tools to achieve a uniform
presentation across all manuals written in
.Nm ,
and to support hyperlinking if supported by the output medium.
.Pp
This reference document describes the structure of manual pages
and the syntax and usage of the
.Nm
language.
The reference implementation of a parsing and formatting tool is
.Xr mandoc 1 ;
the
.Sx COMPATIBILITY
section describes compatibility with other implementations.
.Pp
In an
.Nm
document, lines beginning with the control character
.Sq \&.
are called
.Dq macro lines .
The first word is the macro name.
It consists of two or three letters.
Most macro names begin with a capital letter.
For a list of available macros, see
.Sx MACRO OVERVIEW .
The words following the macro name are arguments to the macro, optionally
including the names of other, callable macros; see
```



```
.Sx MACRO SYNTAX
for details.
.Pp
Lines not beginning with the control character are called
.Dq text lines .
They provide free-form text to be printed; the formatting of the text
depends on the respective processing context:
.Bd -literal -offset indent
\&.Sh Macro lines change control state.
Text lines are interpreted within the current state.
.Ed
.Pp
Many aspects of the basic syntax of the
.Nm
language are based on the
.Xr roff 5
language; see the
.Em LANGUAGE SYNTAX
and
.Em MACRO SYNTAX
sections in the
.Xr roff 5
manual for details, in particular regarding
comments, escape sequences, whitespace, and quoting.
However, using
.Xr roff 5
requests in
.Nm
documents is discouraged;
.Xr mandoc 1
supports some of them merely for backward compatibility.
.Sh MANUAL STRUCTURE
A well-formed
.Nm
document consists of a document prologue followed by one or more
sections.
.Pp
The prologue, which consists of the
.Sx \&Dd ,
.Sx \&Dt ,
and
.Sx \&Os
macros in that order, is required for every document.
.Pp
The first section (sections are denoted by
.Sx \&Sh )
must be the NAME section, consisting of at least one
.Sx \&Nm
followed by
.Sx \&Nd .
.Pp
Following that, convention dictates specifying at least the
.Em SYNOPSIS
and
.Em DESCRIPTION
sections, although this varies between manual sections.
.Pp
The following is a well-formed skeleton
.Nm
file for a utility
.Qq progname :
.Bd -literal -offset indent
\&.Dd Jan 1, 1970
\&.Dt PROGNAME section
\&.Os
\&.Sh NAME
```

```
\&.Nm progname
\&.Nd one line description
\&.\e\(dq .Sh LIBRARY
\&.\e\(dq For sections 2, 3, & 9 only.
\&.Sh SYNOPSIS
\&.Nm progname
\&.Op Fl options
\&.Ar
\&.Sh DESCRIPTION
The
\&.Nm
utility processes files ...
\&.\e\(dq .Sh IMPLEMENTATION NOTES
\&.\e\(dq .Sh RETURN VALUES
\&.\e\(dq For sections 2, 3, & 9 only.
\&.\e\(dq .Sh ENVIRONMENT
\&.\e\(dq For sections 1, 1M, and 5.
\&.\e\(dq For sections 1, 1M, 5, & 6 only.
\&.\e\(dq .Sh FILES
\&.\e\(dq .Sh EXIT STATUS
\&.\e\(dq For sections 1, 1M, and 5.
\&.\e\(dq For sections 1, 1M, & 6 only.
\&.\e\(dq .Sh EXAMPLES
\&.\e\(dq .Sh DIAGNOSTICS
\&.\e\(dq For sections 1, 1M, 5, 6, & 7 only.
\&.\e\(dq .Sh ERRORS
\&.\e\(dq For sections 2, 3, & 9 only.
\&.\e\(dq .Sh ARCHITECTURE
\&.\e\(dq .Sh CODE SET INDEPENDENCE
\&.\e\(dq For sections 1, 1M, & 3 only.
\&.\e\(dq .Sh INTERFACE STABILITY
\&.\e\(dq .Sh MT-LEVEL
\&.\e\(dq For sections 2 & 3 only.
\&.\e\(dq .Sh SECURITY
\&.\e\(dq .Sh SEE ALSO
\&.\e\(dq .Xr foobar 1
\&.\e\(dq .Sh STANDARDS
\&.\e\(dq .Sh HISTORY
\&.\e\(dq .Sh AUTHORS
\&.\e\(dq .Sh CAVEATS
\&.\e\(dq .Sh BUGS
\&.\e\(dq .Sh SECURITY CONSIDERATIONS
\&.\e\(dq Not used in OpenBSD.
.Ed
.Pp
The sections in an
.Nm
document are conventionally ordered as they appear above.
Sections should be composed as follows:
.Bl -ohang -offset Ds
.It Em NAME
The name(s) and a one line description of the documented material.
The syntax for this as follows:
.Bd -literal -offset indent
\&.Nm name0 ,
\&.Nm name1 ,
\&.Nm name2
\&.Nd a one line description
.Ed
.Pp
Multiple
.Sq \&Nm
names should be separated by commas.
.Pp
The
.Sx \&Nm
```



```
macro(s) must precede the
.Sx \&Nd
macro.
.Pp
See
.Sx \&Nm
and
.Sx \&Nd .
.It Em LIBRARY
The name of the library containing the documented material, which is
assumed to be a function in a section 2, 3, or 9 manual.
The syntax for this is as follows:
.Bd -literal -offset indent
\&.Lb libarm
.Ed
.Pp
See
.Sx \&Lb .
.It Em SYNOPSIS
Documents the utility invocation syntax, function call syntax, or device
configuration.
.Pp
For the first, utilities (sections 1 and 1M), this is
For the first, utilities (sections 1, 1M, and 6), this is
generally structured as follows:
.Bd -literal -offset indent
\&.Nm bar
\&.Op Fl v
\&.Op Fl o Ar file
\&.Op Ar
\&.Nm foo
\&.Op Fl v
\&.Op Fl o Ar file
\&.Op Ar
.Ed
.Pp
Commands should be ordered alphabetically.
.Pp
For the second, function calls (sections 2, 3, 7I, 7P, 9):
For the second, function calls (sections 2, 3, 9):
.Bd -literal -offset indent
\&.In header.h
\&.Vt extern const char *global;
\&.Ft "char *"
\&.Fn foo "const char *src"
\&.Ft "char *"
\&.Fn bar "const char *src"
.Ed
.Pp
Ordering of
.Sx \&In ,
.Sx \&Vt ,
.Sx \&Fn ,
and
.Sx \&Fo
macros should follow C header-file conventions.
.Pp
And for the third, configurations (section 7D):
And for the third, configurations (section 7):
.Bd -literal -offset indent
\&.Pa /dev/device_node
\&.Cd \(dqit* at isa? port 0x2e\(dq
\&.Cd \(dqit* at isa? port 0x4e\(dq
.Ed
.Pp
Manuals not in these sections generally don’t need a
```

```
.Em SYNOPSIS .
.Pp
Some macros are displayed differently in the
.Em SYNOPSIS
section, particularly
.Sx \&Nm ,
.Sx \&Cd ,
.Sx \&Fd ,
.Sx \&Fn ,
.Sx \&Fo ,
.Sx \&In ,
.Sx \&Vt ,
and
.Sx \&Ft .
All of these macros are output on their own line.
If two such dissimilar macros are pairwise invoked (except for
.Sx \&Ft
before
.Sx \&Fo
or
.Sx \&Fn ) ,
they are separated by a vertical space, unless in the case of
.Sx \&Fo ,
.Sx \&Fn ,
and
.Sx \&Ft ,
which are always separated by vertical space.
.Pp
When text and macros following an
.Sx \&Nm
macro starting an input line span multiple output lines,
all output lines but the first will be indented to align
with the text immediately following the
.Sx \&Nm
macro, up to the next
.Sx \&Nm ,
.Sx \&Sh ,
or
.Sx \&Ss
macro or the end of an enclosing block, whichever comes first.
.It Em DESCRIPTION
This begins with an expansion of the brief, one line description in
.Em NAME :
.Bd -literal -offset indent
The
\&.Nm
utility does this, that, and the other.
.Ed
.Pp
It usually follows with a breakdown of the options (if documenting a
command), such as:
.Bd -literal -offset indent
The arguments are as follows:
\&.Bl \-tag \-width Ds
\&.It Fl v
Print verbose information.
\&.El
.Ed
.Pp
Manuals not documenting a command won't include the above fragment.
.Pp
Since the
.Em DESCRIPTION
section usually contains most of the text of a manual, longer manuals
often use the
.Sx \&Ss
```



```
macro to form subsections.
In very long manuals, the
.Em DESCRIPTION
may be split into multiple sections, each started by an
.Sx \&Sh
macro followed by a non-standard section name, and each having
several subsections, like in the present
.Nm
manual.
.It Em IMPLEMENTATION NOTES
Implementation-specific notes should be kept here.
This is useful when implementing standard functions that may have side
effects or notable algorithmic implications.
.It Em RETURN VALUES
This section documents the
return values of functions in sections 2, 3, and 9.
.Pp
See
.Sx \&Rv .
.It Em ENVIRONMENT
Lists the environment variables used by the utility,
and explains the syntax and semantics of their values.
The
.Xr environ 5
manual provides examples of typical content and formatting.
.Pp
See
.Sx \&Ev .
.It Em FILES
Documents files used.
It's helpful to document both the file name and a short description of how
the file is used (created, modified, etc.).
.Pp
See
.Sx \&Pa .
.It Em EXIT STATUS
This section documents the
.Sy command exit status for sections 1 and 1M.
.Em command exit status for section 1, 6, and 8 utilities.
Historically, this information was described in
.Em DIAGNOSTICS ,
a practise that is now discouraged.
.Pp
See
.Sx \&Ex .
.It Em EXAMPLES
Example usages.
This often contains snippets of well-formed, well-tested invocations.
Make sure that examples work properly!
.It Em DIAGNOSTICS
.Sy Documents error and diagnostic messages displayed to the user or
.Sy sent to logs. Note that exit
.Sy status and return values should be documented in the
.Em Documents error conditions.
.Em This is most useful in section 4 manuals.
.Em Historically, this section was used in place of
.Em EXIT STATUS
.Sy and
.Sy .Em RETURN VALUES
.Sy sections.
.Em for manuals in sections 1, 6, and 8; however, this practise is
.Em discouraged.
.Pp
See
.Sx \&Bl
.Fl diag .
```

```
.It Em ERRORS
Documents error handling in sections 2, 3, and 9.
.Pp
See
.Sx \&Er .
.It Em ARCHITECTURE
This section is usually absent, but will be present when the
interface is specific to one or more architectures.
.It Em CODE SET INDEPENDENCE
Indicates whether the interface operates correctly with various different
code sets.  True independent code sets will support not only ASCII and
Extended UNIX Codesets (EUC), but also other multi-byte encodings such as
UTF-8 and GB2312.
.Pp
Generally there will be some limitations that are fairly standard.  See
.Xr standards 5 for more information about some of these.  Most interfaces
should support at least UTF-8 in addition to ASCII.
.It Em INTERFACE STABILITY
Indicates the level of commitment to the interface. Interfaces can be described
with in the following ways:
.Bl -tag -width Ds
.It Nm Standard
Indicates that the interface is defined by one or more standards bodies.
Generally, changes to the interface will be carefully managed to conform
to the relevant standards.  These interfaces are generally the most suitable
for use in portable programs.
.It Nm Committed
Indicates that the interface is intended to be preserved for the long-haul, and
will rarely, if ever change, and never without notification (barring
extraordinary and extenuating circumstances). These interfaces are
preferred over other interfaces with the exeception of
.Nm Standard
interfaces.
.It Nm Uncommitted
Indicates that the interface may change.  Generally, changes to these interfaces
should be infrequent, and some effort will be made to address compatibility
considerations when changing or removing such interfaces.  However, there is
no firm commitment to the preservation of the interface.  Most often this
is applied to interfaces where operational experience with the interface
is still limited and some need to change may be anticipated.
.Pp
Consumers should expect to revalidate any
.Nm Uncommitted
interfaces when crossing release boundaries.  Products intended for
use on many releases or intended to support compatibility with future
releases should avoid these interfaces.
.It Nm Volatile
The interface can change at any time for any reason. Often this relates to
interfaces that are part of external software components that are still evolving
rapidly.  Consumers should not expect that the interface (either binary or
source level) will be unchanged from one release to the next.
.It Nm Not-an-Interface
Describes something that is specifically not intended for programmatic
consumption.  For example, specific human-readable output, or the layout
of graphical items on a user interface, may be described this way.  Generally
programmatic alternatives to these will be available, and should be used
when programmatic consumption is needed.
.It Nm Private
This is an internal interface.  Generally these interfaces should only be
used within the project, and should not be used by other programs or modules.
The interface can and will change without notice as the project needs, at
any time.
.Pp
Most often, Private interfaces will lack any documentation whatsoever, and
generally any undocumented interface can be assumed to be Private.
.It Nm Obsolete
```



```
The interface is not intended for use in new projects or programs, and may
be removed at a future date.  The
.Nm Obsolete
word is a modifier that can
be applied to other commitment levels. For example an
.Nm Obsolete Committed
interface is unlikely to be removed or changed, but nonetheless new use
is discouraged (perhaps a better newer alternative is present).
.El
.It Em MT-LEVEL
This section describes considerations for the interface when used within
programs that use multiple threads.  More discussion of these considerations
is made in the MT-Level section of
.Xr attributes 5 .
The interface can be described in the following ways.
.Bl -tag -width Ds
.It Nm Safe
Indicates the interface is safe for use within multiple threads.  There
may be additional caveats that apply, in which case those will be
described.  Note that some interfaces have semantics which may affect
other threads, but these should be an intrinsic part of the interface
rather than an unexpected side effect.  For example, closing a file in
one thread will cause that file to be closed in all threads.
.It Nm Unsafe
Indicates the interface is unsuitable for concurrent use within multiple
threads.  A threaded application may still make use of the interface, but
will be required to provide external synchronization means to ensure that
only a single thread calls the interface at a time.
.It Nm MT-Safe
Indicates that the interface is not only safe for concurrent use, but is
designed for such use.  For example, a
.Nm Safe
interface may make use of a global lock to provide safety, but at reduced
internal concurrency, whereas an
.Nm MT-Safe
interface will be designed to be efficient even when used concurrently.
.It Nm Async-Signal-Safe
Indicates that the library is safe for use within a signal handler.  An
.Nm MT-Safe
interface can be made
.Nm Async-Signal-Safe
by ensuring that it blocks signals when acquiring locks.
.It Nm Safe with Exections
As for
.Nm Safe
but with specific exceptions noted.
.It Nm MT-Safe with Exections
As for
.Nm MT-Safe
but with specific exceptions noted.
.El
.It Em SECURITY
Documents any security precautions that operators should consider.
.It Em SEE ALSO
References other manuals with related topics.
This section should exist for most manuals.
Cross-references should conventionally be ordered first by section, then
alphabetically.
.Pp
References to other documentation concerning the topic of the manual page,
for example authoritative books or journal articles, may also be
provided in this section.
.Pp
See
.Sx \&Rs
and
```

```
.Sx \&Xr .
.It Em STANDARDS
References any standards implemented or used.
If not adhering to any standards, the
.Em HISTORY
section should be used instead.
.Pp
See
.Sx \&St .
.It Em HISTORY
A brief history of the subject, including where it was first implemented,
and when it was ported to or reimplemented for the operating system at hand.
.It Em AUTHORS
Credits to the person or persons who wrote the code and/or documentation.
Authors should generally be noted by both name and email address.
.Pp
See
.Sx \&An .
.It Em CAVEATS
Common misuses and misunderstandings should be explained
in this section.
.It Em BUGS
Known bugs, limitations, and work-arounds should be described
in this section.
.It Em SECURITY CONSIDERATIONS
Documents any security precautions that operators should consider.
.El
.Sh MACRO OVERVIEW
This overview is sorted such that macros of similar purpose are listed
together, to help find the best macro for any given purpose.
Deprecated macros are not included in the overview, but can be found below
in the alphabetical
.Sx MACRO REFERENCE .
.Ss Document preamble and NAME section macros
.Bl -column "Brq, Bro, Brc" description
.It Sx \&Dd Ta document date: Ar month day , year
.It Sx \&Dt Ta document title: Ar TITLE SECTION Op Ar volume | arch
.It Sx \&Os Ta operating system version: Op Ar system Op Ar version
.It Sx \&Nm Ta document name (one argument)
.It Sx \&Nd Ta document description (one line)
.El
.Ss Sections and cross references
.Bl -column "Brq, Bro, Brc" description
.It Sx \&Sh Ta section header (one line)
.It Sx \&Ss Ta subsection header (one line)
.It Sx \&Sx Ta internal cross reference to a section or subsection
.It Sx \&Xr Ta cross reference to another manual page: Ar name section
.It Sx \&Pp , \&Lp Ta start a text paragraph (no arguments)
.El
.Ss Displays and lists
.Bl -column "Brq, Bro, Brc" description
.It Sx \&Bd , \&Ed Ta display block:
.Fl Ar type
.Op Fl offset Ar width
.Op Fl compact
.It Sx \&D1 Ta indented display (one line)
.It Sx \&Dl Ta indented literal display (one line)
.It Sx \&Bl , \&El Ta list block:
.Fl Ar type
.Op Fl width Ar val
.Op Fl offset Ar val
.Op Fl compact
.It Sx \&It Ta list item (syntax depends on Fl Ar type )
.It Sx \&Ta Ta table cell separator in Sx \&Bl Fl column No lists
.It Sx \&Rs , \&%* , \&Re Ta bibliographic block (references)
.El
```



```
.Ss Spacing control
.Bl -column "Brq, Bro, Brc" description
.It Sx \&Pf Ta prefix, no following horizontal space (one argument)
.It Sx \&Ns Ta roman font, no preceding horizontal space (no arguments)
.It Sx \&Ap Ta apostrophe without surrounding whitespace (no arguments)
.It Sx \&Sm Ta switch horizontal spacing mode: Cm on | off
.It Sx \&Bk , \&Ek Ta keep block: Fl words
.It Sx \&br Ta force output line break in text mode (no arguments)
.It Sx \&sp Ta force vertical space: Op Ar height
.El
.Ss Semantic markup for command line utilities:
.Bl -column "Brq, Bro, Brc" description
.It Sx \&Nm Ta start a SYNOPSIS block with the name of a utility
.It Sx \&Fl Ta command line options (flags) (>=0 arguments)
.It Sx \&Cm Ta command modifier (>0 arguments)
.It Sx \&Ar Ta command arguments (>=0 arguments)
.It Sx \&Op , \&Oo , \&Oc Ta optional syntax elements (enclosure)
.It Sx \&Ic Ta internal or interactive command (>0 arguments)
.It Sx \&Ev Ta environmental variable (>0 arguments)
.It Sx \&Pa Ta file system path (>=0 arguments)
.El
.Ss Semantic markup for function libraries:
.Bl -column "Brq, Bro, Brc" description
.It Sx \&Lb Ta function library (one argument)
.It Sx \&In Ta include file (one argument)
.It Sx \&Ft Ta function type (>0 arguments)
.It Sx \&Fo , \&Fc Ta function block: Ar funcname
.It Sx \&Fn Ta function name:
.Op Ar functype
.Ar funcname
.Oo
.Op Ar argtype
.Ar argname
.Oc
.It Sx \&Fa Ta function argument (>0 arguments)
.It Sx \&Vt Ta variable type (>0 arguments)
.It Sx \&Va Ta variable name (>0 arguments)
.It Sx \&Dv Ta defined variable or preprocessor constant (>0 arguments)
.It Sx \&Er Ta error constant (>0 arguments)
.It Sx \&Ev Ta environmental variable (>0 arguments)
.El
.Ss Various semantic markup:
.Bl -column "Brq, Bro, Brc" description
.It Sx \&An Ta author name (>0 arguments)
.It Sx \&Lk Ta hyperlink: Ar uri Op Ar name
.It Sx \&Mt Ta Do mailto Dc hyperlink: Ar address
.It Sx \&Cd Ta kernel configuration declaration (>0 arguments)
.It Sx \&Ad Ta memory address (>0 arguments)
.It Sx \&Ms Ta mathematical symbol (>0 arguments)
.It Sx \&Tn Ta tradename (>0 arguments)
.El
.Ss Physical markup
.Bl -column "Brq, Bro, Brc" description
.It Sx \&Em Ta italic font or underline (emphasis) (>0 arguments)
.It Sx \&Sy Ta boldface font (symbolic) (>0 arguments)
.It Sx \&Li Ta typewriter font (literal) (>0 arguments)
.It Sx \&No Ta return to roman font (normal) (no arguments)
.It Sx \&Bf , \&Ef Ta font block:
.Op Fl Ar type | Cm \&Em | \&Li | \&Sy
.El
.Ss Physical enclosures
.Bl -column "Brq, Bro, Brc" description
.It Sx \&Dq , \&Do , \&Dc Ta enclose in typographic double quotes: Dq text
.It Sx \&Qq , \&Qo , \&Qc Ta enclose in typewriter double quotes: Qq text
.It Sx \&Sq , \&So , \&Sc Ta enclose in single quotes: Sq text
.It Sx \&Ql Ta single-quoted literal text: Ql text
```

```
.It Sx \&Pq , \&Po , \&Pc Ta enclose in parentheses: Pq text
.It Sx \&Bq , \&Bo , \&Bc Ta enclose in square brackets: Bq text
.It Sx \&Brq , \&Bro , \&Brc Ta enclose in curly braces: Brq text
.It Sx \&Aq , \&Ao , \&Ac Ta enclose in angle brackets: Aq text
.It Sx \&Eo , \&Ec Ta generic enclosure
.El
.Ss Text production
.Bl -column "Brq, Bro, Brc" description
.It Sx \&Ex Fl std Ta standard command exit values: Op Ar utility ...
.It Sx \&Rv Fl std Ta standard function return values: Op Ar function ...
.It Sx \&St Ta reference to a standards document (one argument)
.It Sx \&Ux Ta Ux
.It Sx \&At Ta At
.It Sx \&Bx Ta Bx
.It Sx \&Bsx Ta Bsx
.It Sx \&Nx Ta Nx
.It Sx \&Fx Ta Fx
.It Sx \&Ox Ta Ox
.It Sx \&Dx Ta Dx
.El
.Sh MACRO REFERENCE
This section is a canonical reference of all macros, arranged
alphabetically.
For the scoping of individual macros, see
.Sx MACRO SYNTAX .
.Ss \&%A
Author name of an
.Sx \&Rs
block.
Multiple authors should each be accorded their own
.Sx \%%A
line.
Author names should be ordered with full or abbreviated forename(s)
first, then full surname.
.Ss \&%B
Book title of an
.Sx \&Rs
block.
This macro may also be used in a non-bibliographic context when
referring to book titles.
.Ss \&%C
Publication city or location of an
.Sx \&Rs
block.
.Ss \&%D
Publication date of an
.Sx \&Rs
block.
Recommended formats of arguments are
.Ar month day , year
or just
.Ar year .
.Ss \&%I
Publisher or issuer name of an
.Sx \&Rs
block.
.Ss \&%J
Journal name of an
.Sx \&Rs
block.
.Ss \&%N
Issue number (usually for journals) of an
.Sx \&Rs
block.
.Ss \&%O
Optional information of an
```



```
.Sx \&Rs
block.
.Ss \&%P
Book or journal page number of an
.Sx \&Rs
block.
.Ss \&%Q
Institutional author (school, government, etc.) of an
.Sx \&Rs
block.
Multiple institutional authors should each be accorded their own
.Sx \&%Q
line.
.Ss \&%R
Technical report name of an
.Sx \&Rs
block.
.Ss \&%T
Article title of an
.Sx \&Rs
block.
This macro may also be used in a non-bibliographical context when
referring to article titles.
.Ss \&%U
URI of reference document.
.Ss \&%V
Volume number of an
.Sx \&Rs
block.
.Ss \&Ac
Close an
.Sx \&Ao
block.
Does not have any tail arguments.
.Ss \&Ad
Memory address.
Do not use this for postal addresses.
.Pp
Examples:
.Dl \&.Ad [0,$]
.Dl \&.Ad 0x00000000
.Ss \&An
Author name.
Can be used both for the authors of the program, function, or driver
documented in the manual, or for the authors of the manual itself.
Requires either the name of an author or one of the following arguments:
.Pp
.Bl -tag -width "-nosplitX" -offset indent -compact
.It Fl split
Start a new output line before each subsequent invocation of
.Sx \&An .
.It Fl nosplit
The opposite of
.Fl split .
.El
.Pp
The default is
.Fl nosplit .
The effect of selecting either of the
.Fl split
modes ends at the beginning of the
.Em AUTHORS
section.
In the
.Em AUTHORS
section, the default is
```

```
.Fl nosplit
for the first author listing and
.Fl split
for all other author listings.
.Pp
Examples:
.Dl \&.An -nosplit
.Dl \&.An Kristaps Dzonsons \&Aq kristaps@bsd.lv
.Ss \&Ao
Begin a block enclosed by angle brackets.
Does not have any head arguments.
.Pp
Examples:
.Dl \&.Fl -key= \&Ns \&Ao \&Ar val \&Ac
.Pp
See also
.Sx \&Aq .
.Ss \&Ap
Inserts an apostrophe without any surrounding whitespace.
This is generally used as a grammatical device when referring to the verb
form of a function.
.Pp
Examples:
.Dl \&.Fn execve \&Ap d
.Ss \&Aq
Encloses its arguments in angle brackets.
.Pp
Examples:
.Dl \&.Fl -key= \&Ns \&Aq \&Ar val
.Pp
.Em Remarks :
this macro is often abused for rendering URIs, which should instead use
.Sx \&Lk
or
.Sx \&Mt ,
or to note pre-processor
.Dq Li #include
statements, which should use
.Sx \&In .
.Pp
See also
.Sx \&Ao .
.Ss \&Ar
Command arguments.
If an argument is not provided, the string
.Dq file ...\&
is used as a default.
.Pp
Examples:
.Dl ".Fl o Ar file"
.Dl ".Ar"
.Dl ".Ar arg1 , arg2 ."
.Pp
The arguments to the
.Sx \&Ar
macro are names and placeholders for command arguments;
for fixed strings to be passed verbatim as arguments, use
.Sx \&Fl
or
.Sx \&Cm .
.Ss \&At
Formats an AT&T version.
Accepts one optional argument:
.Pp
.Bl -tag -width "v[1-7] | 32vX" -offset indent -compact
.It Cm v[1-7] | 32v
```



```
A version of
.At .
.It Cm III
.At III .
.It Cm V[.[1-4]]?
A version of
.At V .
.El
.Pp
Note that these arguments do not begin with a hyphen.
.Pp
Examples:
.Dl \&.At
.Dl \&.At III
.Dl \&.At V.1
.Pp
See also
.Sx \&Bsx ,
.Sx \&Bx ,
.Sx \&Dx ,
.Sx \&Fx ,
.Sx \&Nx ,
.Sx \&Ox ,
and
.Sx \&Ux .
.Ss \&Bc
Close a
.Sx \&Bo
block.
Does not have any tail arguments.
.Ss \&Bd
Begin a display block.
Its syntax is as follows:
.Bd -ragged -offset indent
.Pf \. Sx \&Bd
.Fl Ns Ar type
.Op Fl offset Ar width
.Op Fl compact
.Ed
.Pp
Display blocks are used to select a different indentation and
justification than the one used by the surrounding text.
They may contain both macro lines and text lines.
By default, a display block is preceded by a vertical space.
.Pp
The
.Ar type
must be one of the following:
.Bl -tag -width 13n -offset indent
.It Fl centered
Produce one output line from each input line, and centre-justify each line.
Using this display type is not recommended; many
.Nm
implementations render it poorly.
.It Fl filled
Change the positions of line breaks to fill each line, and left- and
right-justify the resulting block.
.It Fl literal
Produce one output line from each input line,
and do not justify the block at all.
Preserve white space as it appears in the input.
Always use a constant-width font.
Use this for displaying source code.
.It Fl ragged
Change the positions of line breaks to fill each line, and left-justify
the resulting block.
```

```
.It Fl unfilled
The same as
.Fl literal ,
but using the same font as for normal text, which is a variable width font
if supported by the output device.
.El
.Pp
The
.Ar type
must be provided first.
Additional arguments may follow:
.Bl -tag -width 13n -offset indent
.It Fl offset Ar width
Indent the display by the
.Ar width ,
which may be one of the following:
.Bl -item
.It
One of the pre-defined strings
.Cm indent ,
the width of a standard indentation (six constant width characters);
.Cm indent-two ,
twice
.Cm indent ;
.Cm left ,
which has no effect;
.Cm right ,
which justifies to the right margin; or
.Cm center ,
which aligns around an imagined centre axis.
.It
A macro invocation, which selects a predefined width
associated with that macro.
The most popular is the imaginary macro
.Ar \&Ds ,
which resolves to
.Sy 6n .
.It
A width using the syntax described in
.Sx Scaling Widths .
.It
An arbitrary string, which indents by the length of this string.
.El
.Pp
When the argument is missing,
.Fl offset
is ignored.
.It Fl compact
Do not assert vertical space before the display.
.El
.Pp
Examples:
.Bd -literal -offset indent
\&.Bd \-literal \-offset indent \-compact
   Hello       world.
\&.Ed
.Ed
.Pp
See also
.Sx \&D1
and
.Sx \&Dl .
.Ss \&Bf
Change the font mode for a scoped block of text.
Its syntax is as follows:
.Bd -ragged -offset indent
```



```
.Pf \. Sx \&Bf
.Oo
.Fl emphasis | literal | symbolic |
.Cm \&Em | \&Li | \&Sy
.Oc
.Ed
.Pp
The
.Fl emphasis
and
.Cm \&Em
argument are equivalent, as are
.Fl symbolic
and
.Cm \&Sy ,
and
.Fl literal
and
.Cm \&Li .
Without an argument, this macro does nothing.
The font mode continues until broken by a new font mode in a nested
scope or
.Sx \&Ef
is encountered.
.Pp
See also
.Sx \&Li ,
.Sx \&Ef ,
.Sx \&Em ,
and
.Sx \&Sy .
.Ss \&Bk
For each macro, keep its output together on the same output line,
until the end of the macro or the end of the input line is reached,
whichever comes first.
Line breaks in text lines are unaffected.
The syntax is as follows:
.Pp
.D1 Pf \. Sx \&Bk Fl words
.Pp
The
.Fl words
argument is required; additional arguments are ignored.
.Pp
The following example will not break within each
.Sx \&Op
macro line:
.Bd -literal -offset indent
\&.Bk \-words
\&.Op Fl f Ar flags
\&.Op Fl o Ar output
\&.Ek
.Ed
.Pp
Be careful in using over-long lines within a keep block!
Doing so will clobber the right margin.
.Ss \&Bl
Begin a list.
Lists consist of items specified using the
.Sx \&It
macro, containing a head or a body or both.
The list syntax is as follows:
.Bd -ragged -offset indent
.Pf \. Sx \&Bl
.Fl Ns Ar type
.Op Fl width Ar val
```

```
.Op Fl offset Ar val
.Op Fl compact
.Op HEAD ...
.Ed
.Pp
The list
.Ar type
is mandatory and must be specified first.
The
.Fl width
and
.Fl offset
arguments accept
.Sx Scaling Widths
or use the length of the given string.
The
.Fl offset
is a global indentation for the whole list, affecting both item heads
and bodies.
For those list types supporting it, the
.Fl width
argument requests an additional indentation of item bodies,
to be added to the
.Fl offset .
Unless the
.Fl compact
argument is specified, list entries are separated by vertical space.
.Pp
A list must specify one of the following list types:
.Bl -tag -width 12n -offset indent
.It Fl bullet
No item heads can be specified, but a bullet will be printed at the head
of each item.
Item bodies start on the same output line as the bullet
and are indented according to the
.Fl width
argument.
.It Fl column
A columnated list.
The
.Fl width
argument has no effect; instead, each argument specifies the width
of one column, using either the
.Sx Scaling Widths
syntax or the string length of the argument.
If the first line of the body of a
.Fl column
list is not an
.Sx \&It
macro line,
.Sx \&It
contexts spanning one input line each are implied until an
.Sx \&It
macro line is encountered, at which point items start being interpreted as
described in the
.Sx \&It
documentation.
.It Fl dash
Like
.Fl bullet ,
except that dashes are used in place of bullets.
.It Fl diag
Like
.Fl inset ,
except that item heads are not parsed for macro invocations.
Most often used in the
```



```
.Em DIAGNOSTICS
section with error constants in the item heads.
.It Fl enum
A numbered list.
No item heads can be specified.
Formatted like
.Fl bullet ,
except that cardinal numbers are used in place of bullets,
starting at 1.
.It Fl hang
Like
.Fl tag ,
except that the first lines of item bodies are not indented, but follow
the item heads like in
.Fl inset
lists.
.It Fl hyphen
Synonym for
.Fl dash .
.It Fl inset
Item bodies follow items heads on the same line, using normal inter-word
spacing.
Bodies are not indented, and the
.Fl width
argument is ignored.
.It Fl item
No item heads can be specified, and none are printed.
Bodies are not indented, and the
.Fl width
argument is ignored.
.It Fl ohang
Item bodies start on the line following item heads and are not indented.
The
.Fl width
argument is ignored.
.It Fl tag
Item bodies are indented according to the
.Fl width
argument.
When an item head fits inside the indentation, the item body follows
this head on the same output line.
Otherwise, the body starts on the output line following the head.
.El
.Pp
Lists may be nested within lists and displays.
Nesting of
.Fl column
and
.Fl enum
lists may not be portable.
.Pp
See also
.Sx \&El
and
.Sx \&It .
.Ss \&Bo
Begin a block enclosed by square brackets.
Does not have any head arguments.
.Pp
Examples:
.Bd -literal -offset indent -compact
\&.Bo 1 ,
\&.Dv BUFSIZ \&Bc
.Ed
.Pp
See also
```

```
.Sx \&Bq .
.Ss \&Bq
Encloses its arguments in square brackets.
.Pp
Examples:
.Dl \&.Bq 1 , \&Dv BUFSIZ
.Pp
.Em Remarks :
this macro is sometimes abused to emulate optional arguments for
commands; the correct macros to use for this purpose are
.Sx \&Op ,
.Sx \&Oo ,
and
.Sx \&Oc .
.Pp
See also
.Sx \&Bo .
.Ss \&Brc
Close a
.Sx \&Bro
block.
Does not have any tail arguments.
.Ss \&Bro
Begin a block enclosed by curly braces.
Does not have any head arguments.
.Pp
Examples:
.Bd -literal -offset indent -compact
\&.Bro 1 , ... ,
\&.Va n \&Brc
.Ed
.Pp
See also
.Sx \&Brq .
.Ss \&Brq
Encloses its arguments in curly braces.
.Pp
Examples:
.Dl \&.Brq 1 , ... , \&Va n
.Pp
See also
.Sx \&Bro .
.Ss \&Bsx
Format the BSD/OS version provided as an argument, or a default value if
no argument is provided.
.Pp
Examples:
.Dl \&.Bsx 1.0
.Dl \&.Bsx
.Pp
See also
.Sx \&At ,
.Sx \&Bx ,
.Sx \&Dx ,
.Sx \&Fx ,
.Sx \&Nx ,
.Sx \&Ox ,
and
.Sx \&Ux .
.Ss \&Bt
Prints
.Dq is currently in beta test.
.Ss \&Bx
Format the BSD version provided as an argument, or a default value if no
argument is provided.
.Pp
```



```
Examples:
.Dl \&.Bx 4.3 Tahoe
.Dl \&.Bx 4.4
.Dl \&.Bx
.Pp
See also
.Sx \&At ,
.Sx \&Bsx ,
.Sx \&Dx ,
.Sx \&Fx ,
.Sx \&Nx ,
.Sx \&Ox ,
and
.Sx \&Ux .
.Ss \&Cd
Kernel configuration declaration.  It is found in pages for
.Bx
and not used here.
Kernel configuration declaration.
This denotes strings accepted by
.Xr config 8 .
It is most often used in section 4 manual pages.
.Pp
Examples:
.Dl \&.Cd device le0 at scode?
.Pp
.Em Remarks :
this macro is commonly abused by using quoted literals to retain
whitespace and align consecutive
.Sx \&Cd
declarations.
This practise is discouraged.
.Ss \&Cm
Command modifiers.
Typically used for fixed strings passed as arguments, unless
.Sx \&Fl
is more appropriate.
Also useful when specifying configuration options or keys.
.Pp
Examples:
.Dl ".Nm mt Fl f Ar device Cm rewind"
.Dl ".Nm ps Fl o Cm pid , Ns Cm command"
.Dl ".Nm dd Cm if= Ns Ar file1 Cm of= Ns Ar file2"
.Dl ".Cm IdentityFile Pa ~/.ssh/id_rsa"
.Dl ".Cm LogLevel Dv DEBUG"
.Ss \&D1
One-line indented display.
This is formatted by the default rules and is useful for simple indented
statements.
It is followed by a newline.
.Pp
Examples:
.Dl \&.D1 \&Fl abcdefgh
.Pp
See also
.Sx \&Bd
and
.Sx \&Dl .
.Ss \&Db
Switch debugging mode.
Its syntax is as follows:
.Pp
.D1 Pf \. Sx \&Db Cm on | off
.Pp
This macro is ignored by
.Xr mandoc 1 .
```

```
.Ss \&Dc
Close a
.Sx \&Do
block.
Does not have any tail arguments.
.Ss \&Dd
Document date.
This is the mandatory first macro of any
.Nm
manual.
Its syntax is as follows:
.Pp
.D1 Pf \. Sx \&Dd Ar month day , year
.Pp
The
.Ar month
is the full English month name, the
.Ar day
is an optionally zero-padded numeral, and the
.Ar year
is the full four-digit year.
.Pp
Other arguments are not portable; the
.Xr mandoc 1
utility handles them as follows:
.Bl -dash -offset 3n -compact
.It
To have the date automatically filled in by the
.Ox
version of
.Xr cvs 1 ,
the special string
.Dq $\&Mdocdate$
can be given as an argument.
.It
A few alternative date formats are accepted as well
and converted to the standard form.
.It
If a date string cannot be parsed, it is used verbatim.
.It
If no date string is given, the current date is used.
.El
.Pp
Examples:
.Dl \&.Dd $\&Mdocdate$
.Dl \&.Dd $\&Mdocdate: July 21 2007$
.Dl \&.Dd July 21, 2007
.Pp
See also
.Sx \&Dt
and
.Sx \&Os .
.Ss \&Dl
One-line intended display.
This is formatted as literal text and is useful for commands and
invocations.
It is followed by a newline.
.Pp
Examples:
.Dl \&.Dl % mandoc mdoc.5 \e(ba less
.Pp
See also
.Sx \&Bd
and
.Sx \&D1 .
.Ss \&Do
```



```
Begin a block enclosed by double quotes.
Does not have any head arguments.
.Pp
Examples:
.Bd -literal -offset indent -compact
\&.Do
April is the cruellest month
\&.Dc
\e(em T.S. Eliot
.Ed
.Pp
See also
.Sx \&Dq .
.Ss \&Dq
Encloses its arguments in
.Dq typographic
double-quotes.
.Pp
Examples:
.Bd -literal -offset indent -compact
\&.Dq April is the cruellest month
\e(em T.S. Eliot
.Ed
.Pp
See also
.Sx \&Qq ,
.Sx \&Sq ,
and
.Sx \&Do .
.Ss \&Dt
Document title.
This is the mandatory second macro of any
.Nm
file.
Its syntax is as follows:
.Bd -ragged -offset indent
.Pf \. Sx \&Dt
.Oo
.Ar title
.Oo
.Ar section
.Op Ar volume
.Op Ar arch
.Oc
.Oc
.Ed
.Pp
Its arguments are as follows:
.Bl -tag -width Ds -offset Ds
.It Ar title
The document’s title (name), defaulting to
.Dq UNKNOWN
if unspecified.
It should be capitalised.
.It Ar section
The manual section. It should correspond to the manual’s filename suffix
and defaults to
.Dq 1
if unspecified.
.It Ar volume
This overrides the volume inferred from
.Ar section .
This field is optional.
.It Ar arch
This specifies the machine architecture a manual page applies to,
where relevant.
```

```
.El
.Ss \&Dv
Defined variables such as preprocessor constants, constant symbols,
enumeration values, and so on.
.Pp
Examples:
.Dl \&.Dv NULL
.Dl \&.Dv BUFSIZ
.Dl \&.Dv STDOUT_FILENO
.Pp
See also
.Sx \&Er
and
.Sx \&Ev
for special-purpose constants and
.Sx \&Va
for variable symbols.
.Ss \&Dx
Format the DragonFly BSD version provided as an argument, or a default
value if no argument is provided.
.Pp
Examples:
.Dl \&.Dx 2.4.1
.Dl \&.Dx
.Pp
See also
.Sx \&At ,
.Sx \&Bsx ,
.Sx \&Bx ,
.Sx \&Fx ,
.Sx \&Nx ,
.Sx \&Ox ,
and
.Sx \&Ux .
.Ss \&Ec
Close a scope started by
.Sx \&Eo .
Its syntax is as follows:
.Pp
.D1 Pf \. Sx \&Ec Op Ar TERM
.Pp
The
.Ar TERM
argument is used as the enclosure tail, for example, specifying \e(rq
will emulate
.Sx \&Dc .
.Ss \&Ed
End a display context started by
.Sx \&Bd .
.Ss \&Ef
End a font mode context started by
.Sx \&Bf .
.Ss \&Ek
End a keep context started by
.Sx \&Bk .
.Ss \&El
End a list context started by
.Sx \&Bl .
.Pp
See also
.Sx \&Bl
and
.Sx \&It .
.Ss \&Em
Denotes text that should be
.Em emphasised .
```



```
Note that this is a presentation term and should not be used for
stylistically decorating technical terms.
Depending on the output device, this is usually represented
using an italic font or underlined characters.
.Pp
Examples:
.Dl \&.Em Warnings!
.Dl \&.Em Remarks :
.Pp
See also
.Sx \&Bf ,
.Sx \&Li ,
.Sx \&No ,
and
.Sx \&Sy .
.Ss \&En
This macro is obsolete and not implemented in
.Xr mandoc 1 .
.Ss \&Eo
An arbitrary enclosure.
Its syntax is as follows:
.Pp
.D1 Pf \. Sx \&Eo Op Ar TERM
.Pp
The
.Ar TERM
argument is used as the enclosure head, for example, specifying \e(lq
will emulate
.Sx \&Do .
.Ss \&Er
Error constants for definitions of the
.Va errno
libc global variable.
This is most often used in section 2 and 3 manual pages.
.Pp
Examples:
.Dl \&.Er EPERM
.Dl \&.Er ENOENT
.Pp
See also
.Sx \&Dv
for general constants.
.Ss \&Es
This macro is obsolete and not implemented.
.Ss \&Ev
Environmental variables such as those specified in
.Xr environ 5 .
.Pp
Examples:
.Dl \&.Ev DISPLAY
.Dl \&.Ev PATH
.Pp
See also
.Sx \&Dv
for general constants.
.Ss \&Ex
Insert a standard sentence regarding command exit values of 0 on success
and >0 on failure.
This is most often used in section 1 and 1M manual pages.
This is most often used in section 1, 6, and 8 manual pages.
Its syntax is as follows:
.Pp
.D1 Pf \. Sx \&Ex Fl std Op Ar utility ...
.Pp
If
.Ar utility
```

```
1556 is not specified, the document’s name set by
1557 .Sx \&Nm
1558 is used.
1559 Multiple
1560 .Ar utility
1561 arguments are treated as separate utilities.
1562 .Pp
1563 See also
1564 .Sx \&Rv .
1565 .Ss \&Fa
1566 Function argument.
1567 Its syntax is as follows:
1568 .Bd -ragged -offset indent
1569 .Pf \. Sx \&Fa
1570 .Op Cm argtype
1571 .Cm argname
1572 .Ed
1573 .Pp
1574 This may be invoked for names with or without the corresponding type.
1575 It is also used to specify the field name of a structure.
1576 Most often, the
1577 .Sx \&Fa
1578 macro is used in the
1579 .Em SYNOPSIS
1580 within
1581 .Sx \&Fo
1582 section when documenting multi-line function prototypes.
1583 If invoked with multiple arguments, the arguments are separated by a
1584 comma.
1585 Furthermore, if the following macro is another
1586 .Sx \&Fa ,
1587 the last argument will also have a trailing comma.
1588 .Pp
1589 Examples:
1590 .Dl \&.Fa \(dqconst char *p\(dq
1591 .Dl \&.Fa \(dqint a\(dq \(dqint b\(dq \(dqint c\(dq
1592 .Dl \&.Fa foo
1593 .Pp
1594 See also
1595 .Sx \&Fo .
1596 .Ss \&Fc
1597 End a function context started by
1598 .Sx \&Fo .
1599 .Ss \&Fd
1600 Historically used to document include files.
1601 This usage has been deprecated in favour of
1602 .Sx \&In .
1603 Do not use this macro.
1604 .Pp
1605 See also
1606 .Sx MANUAL STRUCTURE
1607 and
1608 .Sx \&In .
1609 .Ss \&Fl
1610 Command-line flag or option.
1611 Used when listing arguments to command-line utilities.
1612 Prints a fixed-width hyphen
1613 .Sq \-
1614 directly followed by each argument.
1615 If no arguments are provided, a hyphen is printed followed by a space.
1616 If the argument is a macro, a hyphen is prefixed to the subsequent macro
1617 output.
1618 .Pp
1619 Examples:
1620 .Dl ".Fl R Op Fl H | L | P"
1621 .Dl ".Op Fl 1AaCcdFfgHhikLlmnopqRrSsTtux"
```



```
1622 .Dl ".Fl type Cm d Fl name Pa CVS"
1623 .Dl ".Fl Ar signal_number"
1624 .Dl ".Fl o Fl"
1625 .Pp
1626 See also
1627 .Sx \&Cm .
1628 .Ss \&Fn
1629 A function name.
1630 Its syntax is as follows:
1631 .Bd -ragged -offset indent
1632 .Pf \. Ns Sx \&Fn
1633 .Op Ar functype
1634 .Ar funcname
1635 .Op Oo Ar argtype Oc Ar argname
1636 .Ed
1637 .Pp
1638 Function arguments are surrounded in parenthesis and
1639 are delimited by commas.
1640 If no arguments are specified, blank parenthesis are output.
1641 In the
1642 .Em SYNOPSIS
1643 section, this macro starts a new output line,
1644 and a blank line is automatically inserted between function definitions.
1645 .Pp
1646 Examples:
1647 .Dl \&.Fn \(dqint funcname\(dq \(dqint arg0\(dq \(dqint arg1\(dq
1648 .Dl \&.Fn funcname \(dqint arg0\(dq
1649 .Dl \&.Fn funcname arg0
1650 .Pp
1651 .Bd -literal -offset indent -compact
1652 \&.Ft functype
1653 \&.Fn funcname
1654 .Ed
1655 .Pp
1656 When referring to a function documented in another manual page, use
1657 .Sx \&Xr
1658 instead.
1659 See also
1660 .Sx MANUAL STRUCTURE ,
1661 .Sx \&Fo ,
1662 and
1663 .Sx \&Ft .
1664 .Ss \&Fo
1665 Begin a function block.
1666 This is a multi-line version of
1667 .Sx \&Fn .
1668 Its syntax is as follows:
1669 .Pp
1670 .D1 Pf \. Sx \&Fo Ar funcname
1671 .Pp
1672 Invocations usually occur in the following context:
1673 .Bd -ragged -offset indent
1674 .Pf \. Sx \&Ft Ar functype
1675 .br
1676 .Pf \. Sx \&Fo Ar funcname
1677 .br
1678 .Pf \. Sx \&Fa Oo Ar argtype Oc Ar argname
1679 .br
1680 \&.\.\.
1681 .br
1682 .Pf \. Sx \&Fc
1683 .Ed
1684 .Pp
1685 A
1686 .Sx \&Fo
1687 scope is closed by
```

```
.Sx \&Fc .
.Pp
See also
.Sx MANUAL STRUCTURE ,
.Sx \&Fa ,
.Sx \&Fc ,
and
.Sx \&Ft .
.Ss \&Fr
This macro is obsolete and not implemented in
.Xr mandoc 1 .
.Pp
It was used to show function return values.
The syntax was:
.Pp
.Dl Pf . Sx \&Fr Ar value
.Ss \&Ft
A function type.
Its syntax is as follows:
.Pp
.D1 Pf \. Sx \&Ft Ar functype
.Pp
In the
.Em SYNOPSIS
section, a new output line is started after this macro.
.Pp
Examples:
.Dl \&.Ft int
.Bd -literal -offset indent -compact
\&.Ft functype
\&.Fn funcname
.Ed
.Pp
See also
.Sx MANUAL STRUCTURE ,
.Sx \&Fn ,
and
.Sx \&Fo .
.Ss \&Fx
Format the
.Fx
version provided as an argument, or a default value
if no argument is provided.
.Pp
Examples:
.Dl \&.Fx 7.1
.Dl \&.Fx
.Pp
See also
.Sx \&At ,
.Sx \&Bsx ,
.Sx \&Bx ,
.Sx \&Dx ,
.Sx \&Nx ,
.Sx \&Ox ,
and
.Sx \&Ux .
.Ss \&Hf
This macro is not implemented in
.Xr mandoc 1 .
.Pp
It was used to include the contents of a (header) file literally.
The syntax was:
.Pp
.Dl Pf . Sx \&Hf Ar filename
.Ss \&Ic
```



```
Designate an internal or interactive command.
This is similar to
.Sx \&Cm
but used for instructions rather than values.
.Pp
Examples:
.Dl \&.Ic :wq
.Dl \&.Ic hash
.Dl \&.Ic alias
.Pp
Note that using
.Sx \&Bd Fl literal
or
.Sx \&D1
is preferred for displaying code; the
.Sx \&Ic
macro is used when referring to specific instructions.
.Ss \&In
An
.Dq include
file.
When invoked as the first macro on an input line in the
.Em SYNOPSIS
section, the argument is displayed in angle brackets
and preceded by
.Dq #include ,
and a blank line is inserted in front if there is a preceding
function declaration.
This is most often used in section 2, 3, and 9 manual pages.
.Pp
Examples:
.Dl \&.In sys/types.h
.Pp
See also
.Sx MANUAL STRUCTURE .
.Ss \&It
A list item.
The syntax of this macro depends on the list type.
.Pp
Lists
of type
.Fl hang ,
.Fl ohang ,
.Fl inset ,
and
.Fl diag
have the following syntax:
.Pp
.D1 Pf \. Sx \&It Ar args
.Pp
Lists of type
.Fl bullet ,
.Fl dash ,
.Fl enum ,
.Fl hyphen
and
.Fl item
have the following syntax:
.Pp
.D1 Pf \. Sx \&It
.Pp
with subsequent lines interpreted within the scope of the
.Sx \&It
until either a closing
.Sx \&El
or another
```

```
.Sx \&It .
.Pp
The
.Fl tag
list has the following syntax:
.Pp
.D1 Pf \. Sx \&It Op Cm args
.Pp
Subsequent lines are interpreted as with
.Fl bullet
and family.
The line arguments correspond to the list’s left-hand side; body
arguments correspond to the list’s contents.
.Pp
The
.Fl column
list is the most complicated.
Its syntax is as follows:
.Pp
.D1 Pf \. Sx \&It Ar cell Op <TAB> Ar cell ...
.D1 Pf \. Sx \&It Ar cell Op Sx \&Ta Ar cell ...
.Pp
The arguments consist of one or more lines of text and macros
representing a complete table line.
Cells within the line are delimited by tabs or by the special
.Sx \&Ta
block macro.
The tab cell delimiter may only be used within the
.Sx \&It
line itself; on following lines, only the
.Sx \&Ta
macro can be used to delimit cells, and
.Sx \&Ta
is only recognised as a macro when called by other macros,
not as the first macro on a line.
.Pp
Note that quoted strings may span tab-delimited cells on an
.Sx \&It
line.
For example,
.Pp
.Dl .It \(dqcol1 ; <TAB> col2 ;\(dq \&;
.Pp
will preserve the semicolon whitespace except for the last.
.Pp
See also
.Sx \&Bl .
.Ss \&Lb
Specify a library.
The syntax is as follows:
.Pp
.D1 Pf \. Sx \&Lb Ar library
.Pp
The
.Ar library
parameter may be a system library, such as
.Cm libz
or
.Cm libpam ,
in which case a small library description is printed next to the linker
invocation; or a custom library, in which case the library name is
printed in quotes.
This is most commonly used in the
.Em SYNOPSIS
section as described in
.Sx MANUAL STRUCTURE .
```



```
.Pp
Examples:
.Dl \&.Lb libz
.Dl \&.Lb mdoc
.Ss \&Li
Denotes text that should be in a
.Li literal
font mode.
Note that this is a presentation term and should not be used for
stylistically decorating technical terms.
.Pp
On terminal output devices, this is often indistinguishable from
normal text.
.Pp
See also
.Sx \&Bf ,
.Sx \&Em ,
.Sx \&No ,
and
.Sx \&Sy .
.Ss \&Lk
Format a hyperlink.
Its syntax is as follows:
.Pp
.D1 Pf \. Sx \&Lk Ar uri Op Ar name
.Pp
Examples:
.Dl \&.Lk http://bsd.lv \(dqThe BSD.lv Project\(dq
.Dl \&.Lk http://bsd.lv
.Pp
See also
.Sx \&Mt .
.Ss \&Lp
Synonym for
.Sx \&Pp .
.Ss \&Ms
Display a mathematical symbol.
Its syntax is as follows:
.Pp
.D1 Pf \. Sx \&Ms Ar symbol
.Pp
Examples:
.Dl \&.Ms sigma
.Dl \&.Ms aleph
.Ss \&Mt
Format a
.Dq mailto:
hyperlink.
Its syntax is as follows:
.Pp
.D1 Pf \. Sx \&Mt Ar address
.Pp
Examples:
.Dl \&.Mt discuss@manpages.bsd.lv
.Ss \&Nd
A one line description of the manual’s content.
This may only be invoked in the
.Em SYNOPSIS
section subsequent the
.Sx \&Nm
macro.
.Pp
Examples:
.Dl Pf . Sx \&Nd mdoc language reference
.Dl Pf . Sx \&Nd format and display UNIX manuals
.Pp
```

```
1952 The
1953 .Sx \&Nd
1954 macro technically accepts child macros and terminates with a subsequent
1955 .Sx \&Sh
1956 invocation.
1957 Do not assume this behaviour: some
1958 .Xr whatis 1
1959 database generators are not smart enough to parse more than the line
1960 arguments and will display macros verbatim.
1961 .Pp
1962 See also
1963 .Sx \&Nm .
1964 .Ss \&Nm
1965 The name of the manual page, or \(em in particular in section 1
1966 and 1M pages \(em of an additional command or feature documented in
1968 The name of the manual page, or \(em in particular in section 1, 6,
1969 and 8 pages \(em of an additional command or feature documented in
1967 the manual page.
1968 When first invoked, the
1969 .Sx \&Nm
1970 macro expects a single argument, the name of the manual page.
1971 Usually, the first invocation happens in the
1972 .Em NAME
1973 section of the page.
1974 The specified name will be remembered and used whenever the macro is
1975 called again without arguments later in the page.
1976 The
1977 .Sx \&Nm
1978 macro uses
1979 .Sx Block full-implicit
1980 semantics when invoked as the first macro on an input line in the
1981 .Em SYNOPSIS
1982 section; otherwise, it uses ordinary
1983 .Sx In-line
1984 semantics.
1985 .Pp
1986 Examples:
1987 .Bd -literal -offset indent
1988 \&.Sh SYNOPSIS
1989 \&.Nm cat
1990 \&.Op Fl benstuv
1991 \&.Op Ar
1992 .Ed
1993 .Pp
1994 In the
1995 .Em SYNOPSIS
1996 of section 2, 3 and 9 manual pages, use the
1997 .Sx \&Fn
1998 macro rather than
1999 .Sx \&Nm
2000 to mark up the name of the manual page.
2001 .Ss \&No
2002 Normal text.
2003 Closes the scope of any preceding in-line macro.
2004 When used after physical formatting macros like
2005 .Sx \&Em
2006 or
2007 .Sx \&Sy ,
2008 switches back to the standard font face and weight.
2009 Can also be used to embed plain text strings in macro lines
2010 using semantic annotation macros.
2011 .Pp
2012 Examples:
2013 .Dl ".Em italic , Sy bold , No and roman"
2014 .Pp
2015 .Bd -literal -offset indent -compact
```



```
2016 \&.Sm off
2017 \&.Cm :C No / Ar pattern No / Ar replacement No /
2018 \&.Sm on
2019 .Ed
2020 .Pp
2021 See also
2022 .Sx \&Em ,
2023 .Sx \&Li ,
2024 and
2025 .Sx \&Sy .
2026 .Ss \&Ns
2027 Suppress a space between the output of the preceding macro
2028 and the following text or macro.
2029 Following invocation, input is interpreted as normal text
2030 just like after an
2031 .Sx \&No
2032 macro.
2033 .Pp
2034 This has no effect when invoked at the start of a macro line.
2035 .Pp
2036 Examples:
2037 .Dl ".Ar name Ns = Ns Ar value"
2038 .Dl ".Cm :M Ns Ar pattern"
2039 .Dl ".Fl o Ns Ar output"
2040 .Pp
2041 See also
2042 .Sx \&No
2043 and
2044 .Sx \&Sm .
2045 .Ss \&Nx
2046 Format the
2047 .Nx
2048 version provided as an argument, or a default value if
2049 no argument is provided.
2050 .Pp
2051 Examples:
2052 .Dl \&.Nx 5.01
2053 .Dl \&.Nx
2054 .Pp
2055 See also
2056 .Sx \&At ,
2057 .Sx \&Bsx ,
2058 .Sx \&Bx ,
2059 .Sx \&Dx ,
2060 .Sx \&Fx ,
2061 .Sx \&Ox ,
2062 and
2063 .Sx \&Ux .
2064 .Ss \&Oc
2065 Close multi-line
2066 .Sx \&Oo
2067 context.
2068 .Ss \&Oo
2069 Multi-line version of
2070 .Sx \&Op .
2071 .Pp
2072 Examples:
2073 .Bd -literal -offset indent -compact
2074 \&.Oo
2075 \&.Op Fl flag Ns Ar value
2076 \&.Oc
2077 .Ed
2078 .Ss \&Op
2079 Optional part of a command line.
2080 Prints the argument(s) in brackets.
2081 This is most often used in the
```

```
.Em SYNOPSIS
section of section 1 and 1M manual pages.
section of section 1 and 8 manual pages.
.Pp
Examples:
.Dl \&.Op \&Fl a \&Ar b
.Dl \&.Op \&Ar a | b
.Pp
See also
.Sx \&Oo .
.Ss \&Os
Document operating system version.
This is the mandatory third macro of
any
.Nm
file.
Its syntax is as follows:
.Pp
.D1 Pf \. Sx \&Os Op Ar system Op Ar version
.Pp
The optional
.Ar system
parameter specifies the relevant operating system or environment.
Left unspecified, it defaults to the local operating system version.
This is the suggested form.
.Pp
Examples:
.Dl \&.Os
.Dl \&.Os KTH/CSC/TCS
.Dl \&.Os BSD 4.3
.Pp
See also
.Sx \&Dd
and
.Sx \&Dt .
.Ss \&Ot
This macro is obsolete and not implemented in
.Xr mandoc 1 .
.Pp
Historical
.Xr mdoc 5
packages described it as
.Dq "old function type (FORTRAN)" .
.Ss \&Ox
Format the
.Ox
version provided as an argument, or a default value
if no argument is provided.
.Pp
Examples:
.Dl \&.Ox 4.5
.Dl \&.Ox
.Pp
See also
.Sx \&At ,
.Sx \&Bsx ,
.Sx \&Bx ,
.Sx \&Dx ,
.Sx \&Fx ,
.Sx \&Nx ,
and
.Sx \&Ux .
.Ss \&Pa
An absolute or relative file system path, or a file or directory name.
If an argument is not provided, the character
.Sq \(ti
```



```
is used as a default.
.Pp
Examples:
.Dl \&.Pa /usr/bin/mandoc
.Dl \&.Pa /usr/share/man/man5/mdoc.5
.Pp
See also
.Sx \&Lk .
.Ss \&Pc
Close parenthesised context opened by
.Sx \&Po .
.Ss \&Pf
Removes the space between its argument
.Pq Dq prefix
and the following macro.
Its syntax is as follows:
.Pp
.D1 .Pf Ar prefix macro arguments ...
.Pp
This is equivalent to:
.Pp
.D1 .No Ar prefix No \&Ns Ar macro arguments ...
.Pp
Examples:
.Dl ".Pf $ Ar variable_name"
.Dl ".Pf 0x Ar hex_digits"
.Pp
See also
.Sx \&Ns
and
.Sx \&Sm .
.Ss \&Po
Multi-line version of
.Sx \&Pq .
.Ss \&Pp
Break a paragraph.
This will assert vertical space between prior and subsequent macros
and/or text.
.Pp
Paragraph breaks are not needed before or after
.Sx \&Sh
or
.Sx \&Ss
macros or before displays
.Pq Sx \&Bd
or lists
.Pq Sx \&Bl
unless the
.Fl compact
flag is given.
.Ss \&Pq
Parenthesised enclosure.
.Pp
See also
.Sx \&Po .
.Ss \&Qc
Close quoted context opened by
.Sx \&Qo .
.Ss \&Ql
Format a single-quoted literal.
See also
.Sx \&Qq
and
.Sx \&Sq .
.Ss \&Qo
Multi-line version of
```

```
2213 .Sx \&Qq .
2214 .Ss \&Qq
2215 Encloses its arguments in
2216 .Qq typewriter
2217 double-quotes.
2218 Consider using
2219 .Sx \&Dq .
2220 .Pp
2221 See also
2222 .Sx \&Dq ,
2223 .Sx \&Sq ,
2224 and
2225 .Sx \&Qo .
2226 .Ss \&Re
2227 Close an
2228 .Sx \&Rs
2229 block.
2230 Does not have any tail arguments.
2231 .Ss \&Rs
2232 Begin a bibliographic
2233 .Pq Dq reference
2234 block.
2235 Does not have any head arguments.
2236 The block macro may only contain
2237 .Sx \&%A ,
2238 .Sx \&%B ,
2239 .Sx \&%C ,
2240 .Sx \&%D ,
2241 .Sx \&%I ,
2242 .Sx \&%J ,
2243 .Sx \&%N ,
2244 .Sx \&%O ,
2245 .Sx \&%P ,
2246 .Sx \&%Q ,
2247 .Sx \&%R ,
2248 .Sx \&%T ,
2249 .Sx \&%U ,
2250 and
2251 .Sx \&%V
2252 child macros (at least one must be specified).
2253 .Pp
2254 Examples:
2255 .Bd -literal -offset indent -compact
2256 \&.Rs
2257 \&.%A J. E. Hopcroft
2258 \&.%A J. D. Ullman
2259 \&.%B Introduction to Automata Theory, Languages, and Computation
2260 \&.%I Addison-Wesley
2261 \&.%C Reading, Massachusettes
2262 \&.%D 1979
2263 \&.Re
2264 .Ed
2265 .Pp
2266 If an
2267 .Sx \&Rs
2268 block is used within a SEE ALSO section, a vertical space is asserted
2269 before the rendered output, else the block continues on the current
2270 line.
2271 .Ss \&Rv
2272 Insert a standard sentence regarding a function call’s return value of 0
2273 on success and \-1 on error, with the
2274 .Va errno
2275 libc global variable set on error.
2276 Its syntax is as follows:
2277 .Pp
2278 .D1 Pf \. Sx \&Rv Fl std Op Ar function ...
```



```
2279 .Pp
2280 If
2281 .Ar function
2282 is not specified, the document’s name set by
2283 .Sx \&Nm
2284 is used.
2285 Multiple
2286 .Ar function
2287 arguments are treated as separate functions.
2288 .Pp
2289 See also
2290 .Sx \&Ex .
2291 .Ss \&Sc
2292 Close single-quoted context opened by
2293 .Sx \&So .
2294 .Ss \&Sh
2295 Begin a new section.
2296 For a list of conventional manual sections, see
2297 .Sx MANUAL STRUCTURE .
2298 These sections should be used unless it’s absolutely necessary that
2299 custom sections be used.
2300 .Pp
2301 Section names should be unique so that they may be keyed by
2302 .Sx \&Sx .
2303 Although this macro is parsed, it should not consist of child node or it
2304 may not be linked with
2305 .Sx \&Sx .
2306 .Pp
2307 See also
2308 .Sx \&Pp ,
2309 .Sx \&Ss ,
2310 and
2311 .Sx \&Sx .
2312 .Ss \&Sm
2313 Switches the spacing mode for output generated from macros.
2314 Its syntax is as follows:
2315 .Pp
2316 .D1 Pf \. Sx \&Sm Cm on | off
2317 .Pp
2318 By default, spacing is
2319 .Cm on .
2320 When switched
2321 .Cm off ,
2322 no white space is inserted between macro arguments and between the
2323 output generated from adjacent macros, but text lines
2324 still get normal spacing between words and sentences.
2325 .Ss \&So
2326 Multi-line version of
2327 .Sx \&Sq .
2328 .Ss \&Sq
2329 Encloses its arguments in
2330 .Sq typewriter
2331 single-quotes.
2332 .Pp
2333 See also
2334 .Sx \&Dq ,
2335 .Sx \&Qq ,
2336 and
2337 .Sx \&So .
2338 .Ss \&Ss
2339 Begin a new subsection.
2340 Unlike with
2341 .Sx \&Sh ,
2342 there is no convention for the naming of subsections.
2343 Except
2344 .Em DESCRIPTION ,
```

```
2345 the conventional sections described in
2346 .Sx MANUAL STRUCTURE
2347 rarely have subsections.
2348 .Pp
2349 Sub-section names should be unique so that they may be keyed by
2350 .Sx \&Sx .
2351 Although this macro is parsed, it should not consist of child node or it
2352 may not be linked with
2353 .Sx \&Sx .
2354 .Pp
2355 See also
2356 .Sx \&Pp ,
2357 .Sx \&Sh ,
2358 and
2359 .Sx \&Sx .
2360 .Ss \&St
2361 Replace an abbreviation for a standard with the full form.
2362 The following standards are recognised:
2363 .Pp
2364 .Bl -tag -width "-p1003.1g-2000X" -compact
2365 .It \-p1003.1-88
2366 .St -p1003.1-88
2367 .It \-p1003.1-90
2368 .St -p1003.1-90
2369 .It \-p1003.1-96
2370 .St -p1003.1-96
2371 .It \-p1003.1-2001
2372 .St -p1003.1-2001
2373 .It \-p1003.1-2004
2374 .St -p1003.1-2004
2375 .It \-p1003.1-2008
2376 .St -p1003.1-2008
2377 .It \-p1003.1
2378 .St -p1003.1
2379 .It \-p1003.1b
2380 .St -p1003.1b
2381 .It \-p1003.1b-93
2382 .St -p1003.1b-93
2383 .It \-p1003.1c-95
2384 .St -p1003.1c-95
2385 .It \-p1003.1g-2000
2386 .St -p1003.1g-2000
2387 .It \-p1003.1i-95
2388 .St -p1003.1i-95
2389 .It \-p1003.2-92
2390 .St -p1003.2-92
2391 .It \-p1003.2a-92
2392 .St -p1003.2a-92
2393 .It \-p1387.2-95
2394 .St -p1387.2-95
2395 .It \-p1003.2
2396 .St -p1003.2
2397 .It \-p1387.2
2398 .St -p1387.2
2399 .It \-isoC
2400 .St -isoC
2401 .It \-isoC-90
2402 .St -isoC-90
2403 .It \-isoC-amd1
2404 .St -isoC-amd1
2405 .It \-isoC-tcor1
2406 .St -isoC-tcor1
2407 .It \-isoC-tcor2
2408 .St -isoC-tcor2
2409 .It \-isoC-99
2410 .St -isoC-99
```



```
2411 .It \-isoC-2011
2412 .St -isoC-2011
2413 .It \-iso9945-1-90
2414 .St -iso9945-1-90
2415 .It \-iso9945-1-96
2416 .St -iso9945-1-96
2417 .It \-iso9945-2-93
2418 .St -iso9945-2-93
2419 .It \-ansiC
2420 .St -ansiC
2421 .It \-ansiC-89
2422 .St -ansiC-89
2423 .It \-ansiC-99
2424 .St -ansiC-99
2425 .It \-ieee754
2426 .St -ieee754
2427 .It \-iso8802-3
2428 .St -iso8802-3
2429 .It \-iso8601
2430 .St -iso8601
2431 .It \-ieee1275-94
2432 .St -ieee1275-94
2433 .It \-xpg3
2434 .St -xpg3
2435 .It \-xpg4
2436 .St -xpg4
2437 .It \-xpg4.2
2438 .St -xpg4.2
2439 .It \-xpg4.3
2440 .St -xpg4.3
2441 .It \-xbd5
2442 .St -xbd5
2443 .It \-xcu5
2444 .St -xcu5
2445 .It \-xsh5
2446 .St -xsh5
2447 .It \-xns5
2448 .St -xns5
2449 .It \-xns5.2
2450 .St -xns5.2
2451 .It \-xns5.2d2.0
2452 .St -xns5.2d2.0
2453 .It \-xcurses4.2
2454 .St -xcurses4.2
2455 .It \-susv2
2456 .St -susv2
2457 .It \-susv3
2458 .St -susv3
2459 .It \-svid4
2460 .St -svid4
2461 .El
2462 .Ss \&Sx
2463 Reference a section or subsection in the same manual page.
2464 The referenced section or subsection name must be identical to the
2465 enclosed argument, including whitespace.
2466 .Pp
2467 Examples:
2468 .Dl \&.Sx MANUAL STRUCTURE
2469 .Pp
2470 See also
2471 .Sx \&Sh
2472 and
2473 .Sx \&Ss .
2474 .Ss \&Sy
2475 Format enclosed arguments in symbolic
2476 .Pq Dq boldface .
```

```
2477 Note that this is a presentation term and should not be used for
2478 stylistically decorating technical terms.
2479 .Pp
2480 See also
2481 .Sx \&Bf ,
2482 .Sx \&Em ,
2483 .Sx \&Li ,
2484 and
2485 .Sx \&No .
2486 .Ss \&Ta
2487 Table cell separator in
2488 .Sx \&Bl Fl column
2489 lists; can only be used below
2490 .Sx \&It .
2491 .Ss \&Tn
2492 Format a tradename.
2493 .Pp
2494 Since this macro is often implemented to use a small caps font,
2495 it has historically been used for acronyms (like ASCII) as well.
2496 Such usage is not recommended because it would use the same macro
2497 sometimes for semantical annotation, sometimes for physical formatting.
2498 .Pp
2499 Examples:
2500 .Dl \&.Tn IBM
2501 .Ss \&Ud
2502 Prints out
2503 .Dq currently under development.
2504 .Ss \&Ux
2505 Format the UNIX name.
2506 Accepts no argument.
2507 .Pp
2508 Examples:
2509 .Dl \&.Ux
2510 .Pp
2511 See also
2512 .Sx \&At ,
2513 .Sx \&Bsx ,
2514 .Sx \&Bx ,
2515 .Sx \&Dx ,
2516 .Sx \&Fx ,
2517 .Sx \&Nx ,
2518 and
2519 .Sx \&Ox .
2520 .Ss \&Va
2521 A variable name.
2522 .Pp
2523 Examples:
2524 .Dl \&.Va foo
2525 .Dl \&.Va const char *bar ;
2526 .Ss \&Vt
2527 A variable type.
2528 This is also used for indicating global variables in the
2529 .Em SYNOPSIS
2530 section, in which case a variable name is also specified.
2531 Note that it accepts
2532 .Sx Block partial-implicit
2533 syntax when invoked as the first macro on an input line in the
2534 .Em SYNOPSIS
2535 section, else it accepts ordinary
2536 .Sx In-line
2537 syntax.
2538 In the former case, this macro starts a new output line,
2539 and a blank line is inserted in front if there is a preceding
2540 function definition or include directive.
2541 .Pp
2542 Note that this should not be confused with
```



```
2543 .Sx \&Ft ,
2544 which is used for function return types.
2545 .Pp
2546 Examples:
2547 .Dl \&.Vt unsigned char
2548 .Dl \&.Vt extern const char * const sys_signame[] \&;
2549 .Pp
2550 See also
2551 .Sx MANUAL STRUCTURE
2552 and
2553 .Sx \&Va .
2554 .Ss \&Xc
2555 Close a scope opened by
2556 .Sx \&Xo .
2557 .Ss \&Xo
2558 Extend the header of an
2559 .Sx \&It
2560 macro or the body of a partial-implicit block macro
2561 beyond the end of the input line.
2562 This macro originally existed to work around the 9-argument limit
2563 of historic
2564 .Xr roff 5 .
2565 .Ss \&Xr
2566 Link to another manual
2567 .Pq Qq cross-reference .
2568 Its syntax is as follows:
2569 .Pp
2570 .D1 Pf \. Sx \&Xr Ar name section
2571 .Pp
2572 The
2573 .Ar name
2574 and
2575 .Ar section
2576 are the name and section of the linked manual.
2577 If
2578 .Ar section
2579 is followed by non-punctuation, an
2580 .Sx \&Ns
2581 is inserted into the token stream.
2582 This behaviour is for compatibility with
2583 GNU troff.
2584 .Pp
2585 Examples:
2586 .Dl \&.Xr mandoc 1
2587 .Dl \&.Xr mandoc 1 \&;
2588 .Dl \&.Xr mandoc 1 \&Ns s behaviour
2589 .Ss \&br
2590 Emits a line-break.
2591 This macro should not be used; it is implemented for compatibility with
2592 historical manuals.
2593 .Pp
2594 Consider using
2595 .Sx \&Pp
2596 in the event of natural paragraph breaks.
2597 .Ss \&sp
2598 Emits vertical space.
2599 This macro should not be used; it is implemented for compatibility with
2600 historical manuals.
2601 Its syntax is as follows:
2602 .Pp
2603 .D1 Pf \. Sx \&sp Op Ar height
2604 .Pp
2605 The
2606 .Ar height
2607 argument must be formatted as described in
2608 .Sx Scaling Widths .
```

```
2609 If unspecified,
2610 .Sx \&sp
2611 asserts a single vertical space.
2612 .Sh MACRO SYNTAX
2613 The syntax of a macro depends on its classification.
2614 In this section,
2615 .Sq \-arg
2616 refers to macro arguments, which may be followed by zero or more
2617 .Sq parm
2618 parameters;
2619 .Sq \&Yo
2620 opens the scope of a macro; and if specified,
2621 .Sq \&Yc
2622 closes it out.
2623 .Pp
2624 The
2625 .Em Callable
2626 column indicates that the macro may also be called by passing its name
2627 as an argument to another macro.
2628 For example,
2629 .Sq \&.Op \&Fl O \&Ar file
2630 produces
2631 .Sq Op Fl O Ar file .
2632 To prevent a macro call and render the macro name literally,
2633 escape it by prepending a zero-width space,
2634 .Sq \e& .
2635 For example,
2636 .Sq \&Op \e&Fl O
2637 produces
2638 .Sq Op \&Fl O .
2639 If a macro is not callable but its name appears as an argument
2640 to another macro, it is interpreted as opaque text.
2641 For example,
2642 .Sq \&.Fl \&Sh
2643 produces
2644 .Sq Fl \&Sh .
2645 .Pp
2646 The
2647 .Em Parsed
2648 column indicates whether the macro may call other macros by receiving
2649 their names as arguments.
2650 If a macro is not parsed but the name of another macro appears
2651 as an argument, it is interpreted as opaque text.
2652 .Pp
2653 The
2654 .Em Scope
2655 column, if applicable, describes closure rules.
2656 .Ss Block full-explicit
2657 Multi-line scope closed by an explicit closing macro.
2658 All macros contains bodies; only
2659 .Sx \&Bf
2660 and
2661 .Pq optionally
2662 .Sx \&Bl
2663 contain a head.
2664 .Bd -literal -offset indent
2665 \&.Yo \(lB\-arg \(lBparm...\(rB\(rB \(lBhead...\(rB
2666 \(lBbody...\(rB
2667 \&.Yc
2668 .Ed
2669 .Bl -column "MacroX" "CallableX" "ParsedX" "closed by XXX" -offset indent
2670 .It Em Macro Ta Em Callable Ta Em Parsed Ta Em Scope
2671 .It Sx \&Bd  Ta    \&No     Ta    \&No     Ta    closed by Sx \&Ed
2672 .It Sx \&Bf  Ta    \&No     Ta    \&No     Ta    closed by Sx \&Ef
2673 .It Sx \&Bk  Ta    \&No     Ta    \&No     Ta    closed by Sx \&Ek
2674 .It Sx \&Bl  Ta    \&No     Ta    \&No     Ta    closed by Sx \&El
```



```
2675 .It Sx \&Ed  Ta    \&No     Ta    \&No     Ta    opened by Sx \&Bd
2676 .It Sx \&Ef  Ta    \&No     Ta    \&No     Ta    opened by Sx \&Bf
2677 .It Sx \&Ek  Ta    \&No     Ta    \&No     Ta    opened by Sx \&Bk
2678 .It Sx \&El  Ta    \&No     Ta    \&No     Ta    opened by Sx \&Bl
2679 .El
2680 .Ss Block full-implicit
2681 Multi-line scope closed by end-of-file or implicitly by another macro.
2682 All macros have bodies; some
2683 .Po
2684 .Sx \&It Fl bullet ,
2685 .Fl hyphen ,
2686 .Fl dash ,
2687 .Fl enum ,
2688 .Fl item
2689 .Pc
2690 don't have heads; only one
2691 .Po
2692 .Sx \&It
2693 in
2694 .Sx \&Bl Fl column
2695 .Pc
2696 has multiple heads.
2697 .Bd -literal -offset indent
2698 \&.Yo \(lB\-arg \(lBparm...\(rB\(rB \(lBhead... \(lBTa head...\(rB\(rB
2699 \(lBbody...\(rB
2700 .Ed
2701 .Bl -column "MacroX" "CallableX" "ParsedX" "closed by XXXXXXXXXXX" -offset inden
2702 .It Em Macro Ta Em Callable Ta Em Parsed Ta Em Scope
2703 .It Sx \&It Ta \&No Ta Yes  Ta closed by Sx \&It , Sx \&El
2704 .It Sx \&Nd Ta \&No Ta \&No Ta closed by Sx \&Sh
2705 .It Sx \&Nm Ta \&No Ta Yes  Ta closed by Sx \&Nm , Sx \&Sh , Sx \&Ss
2706 .It Sx \&Sh Ta \&No Ta Yes  Ta closed by Sx \&Sh
2707 .It Sx \&Ss Ta \&No Ta Yes  Ta closed by Sx \&Sh , Sx \&Ss
2708 .El
2709 .Pp
2710 Note that the
2711 .Sx \&Nm
2712 macro is a
2713 .Sx Block full-implicit
2714 macro only when invoked as the first macro
2715 in a
2716 .Em SYNOPSIS
2717 section line, else it is
2718 .Sx In-line .
2719 .Ss Block partial-explicit
2720 Like block full-explicit, but also with single-line scope.
2721 Each has at least a body and, in limited circumstances, a head
2722 .Po
2723 .Sx \&Fo ,
2724 .Sx \&Eo
2725 .Pc
2726 and/or tail
2727 .Pq Sx \&Ec .
2728 .Bd -literal -offset indent
2729 \&.Yo \(lB\-arg \(lBparm...\(rB\(rB \(lBhead...\(rB
2730 \(lBbody...\(rB
2731 \&.Yc \(lBtail...\(rB

2733 \&.Yo \(lB\-arg \(lBparm...\(rB\(rB \(lBhead...\(rB \
2734 \(lBbody...\(rB \&Yc \(lBtail...\(rB
2735 .Ed
2736 .Bl -column "MacroX" "CallableX" "ParsedX" "closed by XXXX" -offset indent
2737 .It Em Macro Ta Em Callable Ta Em Parsed Ta Em Scope
2738 .It Sx \&Ac  Ta    Yes      Ta    Yes      Ta    opened by Sx \&Ao
2739 .It Sx \&Ao  Ta    Yes      Ta    Yes      Ta    closed by Sx \&Ac
2740 .It Sx \&Bc  Ta    Yes      Ta    Yes      Ta    closed by Sx \&Bo
```

```
.It Sx \&Bo  Ta    Yes      Ta    Yes      Ta    opened by Sx \&Bc
.It Sx \&Brc Ta    Yes      Ta    Yes      Ta    opened by Sx \&Bro
.It Sx \&Bro Ta    Yes      Ta    Yes      Ta    closed by Sx \&Brc
.It Sx \&Dc  Ta    Yes      Ta    Yes      Ta    opened by Sx \&Do
.It Sx \&Do  Ta    Yes      Ta    Yes      Ta    closed by Sx \&Dc
.It Sx \&Ec  Ta    Yes      Ta    Yes      Ta    opened by Sx \&Eo
.It Sx \&Eo  Ta    Yes      Ta    Yes      Ta    closed by Sx \&Ec
.It Sx \&Fc  Ta    Yes      Ta    Yes      Ta    opened by Sx \&Fo
.It Sx \&Fo  Ta    \&No     Ta    \&No     Ta    closed by Sx \&Fc
.It Sx \&Oc  Ta    Yes      Ta    Yes      Ta    closed by Sx \&Oo
.It Sx \&Oo  Ta    Yes      Ta    Yes      Ta    opened by Sx \&Oc
.It Sx \&Pc  Ta    Yes      Ta    Yes      Ta    closed by Sx \&Po
.It Sx \&Po  Ta    Yes      Ta    Yes      Ta    opened by Sx \&Pc
.It Sx \&Qc  Ta    Yes      Ta    Yes      Ta    opened by Sx \&Oo
.It Sx \&Qo  Ta    Yes      Ta    Yes      Ta    closed by Sx \&Oc
.It Sx \&Re  Ta    \&No     Ta    \&No     Ta    opened by Sx \&Rs
.It Sx \&Rs  Ta    \&No     Ta    \&No     Ta    closed by Sx \&Re
.It Sx \&Sc  Ta    Yes      Ta    Yes      Ta    opened by Sx \&So
.It Sx \&So  Ta    Yes      Ta    Yes      Ta    closed by Sx \&Sc
.It Sx \&Xc  Ta    Yes      Ta    Yes      Ta    opened by Sx \&Xo
.It Sx \&Xo  Ta    Yes      Ta    Yes      Ta    closed by Sx \&Xc
.El
.Ss Block partial-implicit
Like block full-implicit, but with single-line scope closed by the
end of the line.
.Bd -literal -offset indent
\&.Yo \(lB\-arg \(lBval...\(rB\(rB \(lBbody...\(rB \(lBres...\(rB
.Ed
.Bl -column "MacroX" "CallableX" "ParsedX" -offset indent
.It Em Macro Ta Em Callable Ta Em Parsed
.It Sx \&Aq  Ta    Yes      Ta    Yes
.It Sx \&Bq  Ta    Yes      Ta    Yes
.It Sx \&Brq Ta    Yes      Ta    Yes
.It Sx \&D1  Ta    \&No     Ta    \&Yes
.It Sx \&Dl  Ta    \&No     Ta    Yes
.It Sx \&Dq  Ta    Yes      Ta    Yes
.It Sx \&Op  Ta    Yes      Ta    Yes
.It Sx \&Pq  Ta    Yes      Ta    Yes
.It Sx \&Ql  Ta    Yes      Ta    Yes
.It Sx \&Qq  Ta    Yes      Ta    Yes
.It Sx \&Sq  Ta    Yes      Ta    Yes
.It Sx \&Vt  Ta    Yes      Ta    Yes
.El
.Pp
Note that the
.Sx \&Vt
macro is a
.Sx Block partial-implicit
only when invoked as the first macro
in a
.Em SYNOPSIS
section line, else it is
.Sx In-line .
.Ss Special block macro
The
.Sx \&Ta
macro can only be used below
.Sx \&It
in
.Sx \&Bl Fl column
lists.
It delimits blocks representing table cells;
these blocks have bodies, but no heads.
.Bl -column "MacroX" "CallableX" "ParsedX" "closed by XXXX" -offset indent
.It Em Macro Ta Em Callable Ta Em Parsed Ta Em Scope
.It Sx \&Ta  Ta    Yes      Ta    Yes    Ta closed by Sx \&Ta , Sx \&It
```



```
.El
.Ss In-line
Closed by the end of the line, fixed argument lengths,
and/or subsequent macros.
In-line macros have only text children.
If a number (or inequality) of arguments is
.Pq n ,
then the macro accepts an arbitrary number of arguments.
.Bd -literal -offset indent
\&.Yo \(lB\-arg \(lBval...\(rB\(rB \(lBargs...\(rB \(lBres...\(rB

\&.Yo \(lB\-arg \(lBval...\(rB\(rB \(lBargs...\(rB Yc...

\&.Yo \(lB\-arg \(lBval...\(rB\(rB arg0 arg1 argN
.Ed
.Bl -column "MacroX" "CallableX" "ParsedX" "Arguments" -offset indent
.It Em Macro Ta Em Callable Ta Em Parsed Ta Em Arguments
.It Sx \&%A  Ta    \&No     Ta    \&No     Ta    >0
.It Sx \&%B  Ta    \&No     Ta    \&No     Ta    >0
.It Sx \&%C  Ta    \&No     Ta    \&No     Ta    >0
.It Sx \&%D  Ta    \&No     Ta    \&No     Ta    >0
.It Sx \&%I  Ta    \&No     Ta    \&No     Ta    >0
.It Sx \&%J  Ta    \&No     Ta    \&No     Ta    >0
.It Sx \&%N  Ta    \&No     Ta    \&No     Ta    >0
.It Sx \&%O  Ta    \&No     Ta    \&No     Ta    >0
.It Sx \&%P  Ta    \&No     Ta    \&No     Ta    >0
.It Sx \&%Q  Ta    \&No     Ta    \&No     Ta    >0
.It Sx \&%R  Ta    \&No     Ta    \&No     Ta    >0
.It Sx \&%T  Ta    \&No     Ta    \&No     Ta    >0
.It Sx \&%U  Ta    \&No     Ta    \&No     Ta    >0
.It Sx \&%V  Ta    \&No     Ta    \&No     Ta    >0
.It Sx \&Ad  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&An  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&Ap  Ta    Yes      Ta    Yes      Ta    0
.It Sx \&Ar  Ta    Yes      Ta    Yes      Ta    n
.It Sx \&At  Ta    Yes      Ta    Yes      Ta    1
.It Sx \&Bsx Ta    Yes      Ta    Yes      Ta    n
.It Sx \&Bt  Ta    \&No     Ta    \&No     Ta    0
.It Sx \&Bx  Ta    Yes      Ta    Yes      Ta    n
.It Sx \&Cd  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&Cm  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&Db  Ta    \&No     Ta    \&No     Ta    1
.It Sx \&Dd  Ta    \&No     Ta    \&No     Ta    n
.It Sx \&Dt  Ta    \&No     Ta    \&No     Ta    n
.It Sx \&Dv  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&Dx  Ta    Yes      Ta    Yes      Ta    n
.It Sx \&Em  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&En  Ta    \&No     Ta    \&No     Ta    0
.It Sx \&Er  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&Es  Ta    \&No     Ta    \&No     Ta    0
.It Sx \&Ev  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&Ex  Ta    \&No     Ta    \&No     Ta    n
.It Sx \&Fa  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&Fd  Ta    \&No     Ta    \&No     Ta    >0
.It Sx \&Fl  Ta    Yes      Ta    Yes      Ta    n
.It Sx \&Fn  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&Fr  Ta    \&No     Ta    \&No     Ta    n
.It Sx \&Ft  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&Fx  Ta    Yes      Ta    Yes      Ta    n
.It Sx \&Hf  Ta    \&No     Ta    \&No     Ta    n
.It Sx \&Ic  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&In  Ta    \&No     Ta    \&No     Ta    1
.It Sx \&Lb  Ta    \&No     Ta    \&No     Ta    1
.It Sx \&Li  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&Lk  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&Lp  Ta    \&No     Ta    \&No     Ta    0
```

```
.It Sx \&Ms  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&Mt  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&Nm  Ta    Yes      Ta    Yes      Ta    n
.It Sx \&No  Ta    Yes      Ta    Yes      Ta    0
.It Sx \&Ns  Ta    Yes      Ta    Yes      Ta    0
.It Sx \&Nx  Ta    Yes      Ta    Yes      Ta    n
.It Sx \&Os  Ta    \&No     Ta    \&No     Ta    n
.It Sx \&Ot  Ta    \&No     Ta    \&No     Ta    n
.It Sx \&Ox  Ta    Yes      Ta    Yes      Ta    n
.It Sx \&Pa  Ta    Yes      Ta    Yes      Ta    n
.It Sx \&Pf  Ta    Yes      Ta    Yes      Ta    1
.It Sx \&Pp  Ta    \&No     Ta    \&No     Ta    0
.It Sx \&Rv  Ta    \&No     Ta    \&No     Ta    n
.It Sx \&Sm  Ta    \&No     Ta    \&No     Ta    1
.It Sx \&St  Ta    \&No     Ta    Yes      Ta    1
.It Sx \&Sx  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&Sy  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&Tn  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&Ud  Ta    \&No     Ta    \&No     Ta    0
.It Sx \&Ux  Ta    Yes      Ta    Yes      Ta    n
.It Sx \&Va  Ta    Yes      Ta    Yes      Ta    n
.It Sx \&Vt  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&Xr  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&br  Ta    \&No     Ta    \&No     Ta    0
.It Sx \&sp  Ta    \&No     Ta    \&No     Ta    1
.El
.Ss Delimiters
When a macro argument consists of one single input character
considered as a delimiter, the argument gets special handling.
This does not apply when delimiters appear in arguments containing
more than one character.
Consequently, to prevent special handling and just handle it
like any other argument, a delimiter can be escaped by prepending
a zero-width space
.Pq Sq \e& .
In text lines, delimiters never need escaping, but may be used
as normal punctuation.
.Pp
For many macros, when the leading arguments are opening delimiters,
these delimiters are put before the macro scope,
and when the trailing arguments are closing delimiters,
these delimiters are put after the macro scope.
For example,
.Pp
.D1 Pf \. \&Aq "( [ word ] ) ."
.Pp
renders as:
.Pp
.D1 Aq ( [ word ] ) .
.Pp
Opening delimiters are:
.Pp
.Bl -tag -width Ds -offset indent -compact
.It \&(
left parenthesis
.It \&[
left bracket
.El
.Pp
Closing delimiters are:
.Pp
.Bl -tag -width Ds -offset indent -compact
.It \&.
period
.It \&,
comma
```



```
.It \&:
colon
.It \&;
semicolon
.It \&)
right parenthesis
.It \&]
right bracket
.It \&?
question mark
.It \&!
exclamation mark
.El
.Pp
Note that even a period preceded by a backslash
.Pq Sq \e.\&
gets this special handling; use
.Sq \e&.
to prevent that.
.Pp
Many in-line macros interrupt their scope when they encounter
delimiters, and resume their scope when more arguments follow that
are not delimiters.
For example,
.Pp
.D1 Pf \. \&Fl "a ( b | c \e*(Ba d ) e"
.Pp
renders as:
.Pp
.D1 Fl a ( b | c \*(Ba d ) e
.Pp
This applies to both opening and closing delimiters,
and also to the middle delimiter:
.Pp
.Bl -tag -width Ds -offset indent -compact
.It \&|
vertical bar
.El
.Pp
As a special case, the predefined string \e*(Ba is handled and rendered
in the same way as a plain
.Sq \&|
character.
Using this predefined string is not recommended in new manuals.
.Ss Font handling
In
.Nm
documents, usage of semantic markup is recommended in order to have
proper fonts automatically selected; only when no fitting semantic markup
is available, consider falling back to
.Sx Physical markup
macros.
Whenever any
.Nm
macro switches the
.Xr roff 5
font mode, it will automatically restore the previous font when exiting
its scope.
Manually switching the font using the
.Xr roff 5
.Ql \ef
font escape sequences is never required.
.Sh COMPATIBILITY
This section documents compatibility between mandoc and other other
troff implementations, at this time limited to GNU troff
.Pq Qq groff .
```

```
3005 The term
3006 .Qq historic groff
3007 refers to groff versions before 1.17,
3008 which featured a significant update of the
3009 .Pa doc.tmac
3010 file.
3011 .Pp
3012 Heirloom troff, the other significant troff implementation accepting
3013 \-mdoc, is similar to historic groff.
3014 .Pp
3015 The following problematic behaviour is found in groff:
3016 .ds hist (Historic groff only.)
3017 .Pp
3018 .Bl -dash -compact
3019 .It
3020 Display macros
3021 .Po
3022 .Sx \&Bd ,
3023 .Sx \&Dl ,
3024 and
3025 .Sx \&D1
3026 .Pc
3027 may not be nested.
3028 \*[hist]
3029 .It
3030 .Sx \&At
3031 with unknown arguments produces no output at all.
3032 \*[hist]
3033 Newer groff and mandoc print
3034 .Qq AT&T UNIX
3035 and the arguments.
3036 .It
3037 .Sx \&Bl Fl column
3038 does not recognise trailing punctuation characters when they immediately
3039 precede tabulator characters, but treats them as normal text and
3040 outputs a space before them.
3041 .It
3042 .Sx \&Bd Fl ragged compact
3043 does not start a new line.
3044 \*[hist]
3045 .It
3046 .Sx \&Dd
3047 with non-standard arguments behaves very strangely.
3048 When there are three arguments, they are printed verbatim.
3049 Any other number of arguments is replaced by the current date,
3050 but without any arguments the string
3051 .Dq Epoch
3052 is printed.
3053 .It
3054 .Sx \&Fl
3055 does not print a dash for an empty argument.
3056 \*[hist]
3057 .It
3058 .Sx \&Fn
3059 does not start a new line unless invoked as the line macro in the
3060 .Em SYNOPSIS
3061 section.
3062 \*[hist]
3063 .It
3064 .Sx \&Fo
3065 with
3066 .Pf non- Sx \&Fa
3067 children causes inconsistent spacing between arguments.
3068 In mandoc, a single space is always inserted between arguments.
3069 .It
3070 .Sx \&Ft
```



```
3071 in the
3072 .Em SYNOPSIS
3073 causes inconsistent vertical spacing, depending on whether a prior
3074 .Sx \&Fn
3075 has been invoked.
3076 See
3077 .Sx \&Ft
3078 and
3079 .Sx \&Fn
3080 for the normalised behaviour in mandoc.
3081 .It
3082 .Sx \&In
3083 ignores additional arguments and is not treated specially in the
3084 .Em SYNOPSIS .
3085 \*[hist]
3086 .It
3087 .Sx \&It
3088 sometimes requires a
3089 .Fl nested
3090 flag.
3091 \*[hist]
3092 In new groff and mandoc, any list may be nested by default and
3093 .Fl enum
3094 lists will restart the sequence only for the sub-list.
3095 .It
3096 .Sx \&Li
3097 followed by a delimiter is incorrectly used in some manuals
3098 instead of properly quoting that character, which sometimes works with
3099 historic groff.
3100 .It
3101 .Sx \&Lk
3102 only accepts a single link-name argument; the remainder is misformatted.
3103 .It
3104 .Sx \&Pa
3105 does not format its arguments when used in the FILES section under
3106 certain list types.
3107 .It
3108 .Sx \&Ta
3109 can only be called by other macros, but not at the beginning of a line.
3110 .It
3111 .Sx \&%C
3112 is not implemented.
3113 .It
3114 Historic groff only allows up to eight or nine arguments per macro input
3115 line, depending on the exact situation.
3116 Providing more arguments causes garbled output.
3117 The number of arguments on one input line is not limited with mandoc.
3118 .It
3119 Historic groff has many un-callable macros.
3120 Most of these (excluding some block-level macros) are callable
3121 in new groff and mandoc.
3122 .It
3123 .Sq \(ba
3124 (vertical bar) is not fully supported as a delimiter.
3125 \*[hist]
3126 .It
3127 .Sq \ef
3128 .Pq font face
3129 and
3130 .Sq \ef
3131 .Pq font family face
3132 .Sx Text Decoration
3133 escapes behave irregularly when specified within line-macro scopes.
3134 .It
3135 Negative scaling units return to prior lines.
3136 Instead, mandoc truncates them to zero.
```

```
.El
.Pp
The following features are unimplemented in mandoc:
.Pp
.Bl -dash -compact
.It
.Sx \&Bd
.Fl file Ar file .
.It
.Sx \&Bd
.Fl offset Ar center
and
.Fl offset Ar right .
Groff does not implement centred and flush-right rendering either,
but produces large indentations.
.It
The
.Sq \eh
.Pq horizontal position ,
.Sq \ev
.Pq vertical position ,
.Sq \em
.Pq text colour ,
.Sq \eM
.Pq text filling colour ,
.Sq \ez
.Pq zero-length character ,
.Sq \ew
.Pq string length ,
.Sq \ek
.Pq horizontal position marker ,
.Sq \eo
.Pq text overstrike ,
and
.Sq \es
.Pq text size
escape sequences are all discarded in mandoc.
.It
The
.Sq \ef
scaling unit is accepted by mandoc, but rendered as the default unit.
.It
In quoted literals, groff allows pairwise double-quotes to produce a
standalone double-quote in formatted output.
This is not supported by mandoc.
.El
.Sh SEE ALSO
.Xr man 1 ,
.Xr mandoc 1 ,
.Xr eqn 5 ,
.Xr man 5 ,
.Xr mandoc_char 5 ,
.Xr roff 5 ,
.Xr tbl 5
.Sh HISTORY
The
.Nm
language first appeared as a troff macro package in
.Bx 4.4 .
It was later significantly updated by Werner Lemberg and Ruslan Ermilov
in groff-1.17.
The standalone implementation that is part of the
.Xr mandoc 1
utility written by Kristaps Dzonsons appeared in
.Ox 4.6 .
in July, 2014.
```



```
.Sh AUTHORS
The
.Nm
reference was written by
.An Kristaps Dzonsons ,
.Mt kristaps@bsd.lv .
```