```
(RPAQQ MAINTAINCOMS
        ((FNS MAINTAIN \GETMAINTAINCOMMAND)
         (FNS \MT.ADD.FRIEND \MT.ADD.MEMBER \MT.ADD.OWNER \MT.CHANGE.PASSWORD \MT.CHANGE.REMARK
              \MT.CHANGE.GROUP.COMPONENT \MT.LIST.GROUPS \MT.LOGIN \MT.REMOVE.FRIEND \MT.REMOVE.MEMBER
              \MT.REMOVE.OWNER \MT.TYPE.ENTRY \MT.TYPE.MEMBERS)
         (FNS \MT.MAYBE.PRINT.OVERSTAMPED.RLIST \MT.MAYBE.PRINT.STRING \MT.PRINTRLIST \MT.PRINTSTRINGLIST
              \SKIPCOMPONENT \MT.SKIPSTRINGLIST \MT.RECEIVE.ENTRY)
         (FNS \MT.READRNAME \MT.PERMIT.NS)
         (VARS \MT.ELLIPSIS)
         (DECLARE%: EVAL@COMPILE DONTCOPY (CONSTANTS * GVNAMETYPES)
                (MACROS .ELLIPSIS.)
                (GLOBALVARS \MT.ELLIPSIS))
         (FILES GRAPEVINE)))

(DEFINEQ
```

(\MAINTAIN
```
  [LAMBDA NIL                                          (* bvm%: "20-Jul-85 17:58")
    (PROG (GVUSERNAME GVPASSWORD GVGROUPS LASTNAME LASTGROUP LASTSTRING RNAMEDELIMITERS)
          (DECLARE (SPECVARS GVUSERNAME GVPASSWORD GVGROUPS LASTNAME LASTGROUP LASTSTRING))
          (\MT.LOGIN T)
          (repeatwhile (NULL (ERSETQ (bind CMD while (SETQ CMD (\GETMAINTAINCOMMAND)) do (APPLY* CMD])
```

(\GETMAINTAINCOMMAND
```
  [LAMBDA NIL                                          (* bvm%: "20-Jul-85 17:56")
    (TERPRI T)                                         (* Unimplemented commands%: ("List All Groups" "" RETURN
                                                       (FUNCTION \MT.LIST.GROUPS)))
    (ASKUSER NIL NIL "GV: " '[("Add Friend" "" RETURN (FUNCTION \MT.ADD.FRIEND))
                             ("Add Member" "" RETURN (FUNCTION \MT.ADD.MEMBER))
                             ("Add Owner" "" RETURN (FUNCTION \MT.ADD.OWNER))
                             ("Change Password" "" RETURN (FUNCTION \MT.CHANGE.PASSWORD))
                             ("Change Remark" "" RETURN (FUNCTION \MT.CHANGE.REMARK))
                             ("Login" "" RETURN (FUNCTION \MT.LOGIN))
                             ("Quit" " [confirm]" CONFIRMFLG T RETURN NIL)
                             ("Permit Pseudo-NS names (must type CR to terminate names) [confirm]"
                              "" CONFIRMFLG T RETURN (FUNCTION \MT.PERMIT.NS))
                             ("Remove Friend" "" RETURN (FUNCTION \MT.REMOVE.FRIEND))
                             ("Remove Member" "" RETURN (FUNCTION \MT.REMOVE.MEMBER))
                             ("Remove Owner" "" RETURN (FUNCTION \MT.REMOVE.OWNER))
                             ("Type Entry" "" RETURN (FUNCTION \MT.TYPE.ENTRY))
                             ("Type Members" "" RETURN (FUNCTION \MT.TYPE.MEMBERS))
                             (%Ü "^Y - Enter Lisp" NOECHOFLG T RETURN (FUNCTION (LAMBDA NIL
                                                                                 (TERPRI T)
                                                                                 (USEREXEC '__]
          T NIL '(AUTOCOMPLETEFLG T])

)

(DEFINEQ
```

(\MT.ADD.FRIEND
```
  [LAMBDA NIL                                          (* bvm%: "17-SEP-83 14:16")
    (\MT.CHANGE.GROUP.COMPONENT (FUNCTION GV.ADDFRIEND)
           'ADD])
```

(\MT.ADD.MEMBER
```
  [LAMBDA NIL                                          (* bvm%: "17-SEP-83 14:16")
    (\MT.CHANGE.GROUP.COMPONENT (FUNCTION GV.ADDMEMBER)
           'ADD])
```

(\MT.ADD.OWNER
```
  [LAMBDA NIL                                          (* bvm%: "26-Apr-84 10:44")
    (\MT.CHANGE.GROUP.COMPONENT (FUNCTION GV.ADDOWNER)
           'ADD])
```

(\MT.CHANGE.PASSWORD

```
  [LAMBDA NIL                                                      (* bvm%: "17-NOV-83 14:45")
    (PROG (NAME UPNAME PASS PASSKEY RESULT)
          (DECLARE (USEDFREE GVUSERNAME GVPASSWORD LASTNAME LASTSTRING))
          (COND
             ([SETQ NAME (\MT.READRNAME " for individual: " (CONCAT (CAR GVUSERNAME)
                                                                     "."
                                                                     (CDR GVUSERNAME]
                (COND
                   ([OR [NULL (SETQ PASS (PROMPTFORWORD " to be: " NIL NIL T '*]
                        (NOT (STREQUAL PASS (PROMPTFORWORD " (retype password) " NIL NIL T '*]
                     (printout T " xxx" T))
                   (T (.ELLIPSIS.)
                      (COND
                         ((EQ (SETQ RESULT (GV.CHANGEPASSWORD (SETQ UPNAME (\CHECKNAME NAME))
                                                             (SETQ PASSKEY (GV.MAKEKEY PASS T))
                                                             GVUSERNAME GVPASSWORD))
                              T)
                           (printout T " done" T)
                           (SETQ GVUSERNAME UPNAME)
                           (SETQ GVPASSWORD PASSKEY)
                           (SETPASSWORD 'GV (MKATOM (SETQ LASTNAME (SETQ LASTSTRING NAME)))
                                       PASS))
                         (T (printout T RESULT])
```

⟨\MT.CHANGE.REMARK

```
  [LAMBDA NIL                                                      (* bvm%: "17-SEP-83 15:23")
    (PROG (GVNAMETYPE GROUP RESULT NEWREMARK)
          (DECLARE (SPECVARS GVNAMETYPE)
                   (USEDFREE LASTGROUP LASTSTRING GVUSERNAME GVPASSWORD))
          (COND
             ((SETQ GROUP (\MT.READRNAME " for group: " LASTGROUP))
              (.ELLIPSIS.)
              [COND
                 ((STRINGP (SETQ RESULT (GV.READREMARK GROUP)))
                  (printout T " to be (type remark, terminate with <cr>)" T)
                  (COND
                     ([SETQ NEWREMARK (PROMPTFORWORD "Remark: " RESULT NIL T NIL NIL (CHARCODE (CR]
                      (.ELLIPSIS.)
                      (SETQ RESULT (GV.CHANGEREMARK GROUP NEWREMARK GVUSERNAME GVPASSWORD)))
                     (T (RETURN]
              (printout T (COND
                             ((EQ RESULT T)
                              "done")
                             (T RESULT))
                        T)
              (SETQ LASTSTRING (SETQ LASTGROUP GROUP])
```

⟨\MT.CHANGE.GROUP.COMPONENT

```
  [LAMBDA (GVACCESSFN OPERATION)                                   (* bvm%: "16-SEP-83 23:05")
    (PROG (GVNAMETYPE GROUP INDIVIDUAL RESULT)
          (DECLARE (SPECVARS GVNAMETYPE)
                   (USEDFREE LASTNAME LASTGROUP LASTSTRING GVUSERNAME GVPASSWORD))
          (COND
             ((AND (SETQ INDIVIDUAL (\MT.READRNAME " name: " LASTNAME))
                   (SETQ GROUP (\MT.READRNAME (SELECTQ OPERATION
                                                       (ADD " to group: ")
                                                       (REMOVE " from group: ")
                                                       (SHOULDNT))
                                             LASTGROUP)))
              (.ELLIPSIS.)
              (SETQ RESULT (APPLY* GVACCESSFN GROUP INDIVIDUAL GVUSERNAME GVPASSWORD))
              (printout T (COND
                             ((EQ RESULT T)
                              "done")
                             (T RESULT))
                        T)
              (SETQ LASTNAME INDIVIDUAL)
              (SETQ LASTSTRING (SETQ LASTGROUP GROUP])
```

⟨\MT.LIST.GROUPS

```
  [LAMBDA NIL                                                      (* bvm%: "17-SEP-83 15:52")
    (PROG (GVNAMETYPE NAME GROUPS REG FOUNDONE)
          (DECLARE (SPECVARS GVNAMETYPE)
                   (USEDFREE LASTNAME LASTGROUP LASTSTRING GVGROUPS))
          (COND
             ((AND (SETQ REG (PROMPTFORWORD " in registry: " DEFAULTREGISTRY))
                   (SETQ NAME (\MT.READRNAME " that contain the name: " LASTNAME)))
              (printout T " ... enumerating groups")
              (COND
                 ((EQ (SETQ GROUPS (GV.READMEMBERS (CONS 'GROUPS (MKATOM REG))
                                                   (CAR GVGROUPS)))
                      'NoChange)
                  (SETQ GROUPS GVGROUPS))
                 (T (SETQ GVGROUPS GROUPS)))
```

```
                        (printout T " done." T)
                        (for GROUP in (CDR GROUPS) when (GV.ISMEMBERCLOSURE GROUP NAME)
                           do (COND
                                  (FOUNDONE (printout T ", "))
                                  (T (SETQ FOUNDONE T)))
                              (PRIN1 GROUP T))
                        (SETQ LASTSTRING (SETQ LASTNAME NAME])
```

(**MT.LOGIN**
```
  [LAMBDA (QUIET)                                          (* bvm%: "17-SEP-83 14:18")
    (DECLARE (USEDFREE GVUSERNAME GVPASSWORD LASTNAME LASTSTRING))
    (PROG ((ALWAYSASK (NULL QUIET))
            LOGINFO UPNAME PASSKEY EC)
      LP  (COND
              ((NOT (SETQ LOGINFO (\INTERNAL/GETPASSWORD 'GV ALWAYSASK)))
               (RETURN)))
          (SETQ UPNAME (\CHECKNAME (CAR LOGINFO)))
          (COND
              ((EQ [SETQ EC (GV.AUTHENTICATE UPNAME (SETQ PASSKEY (GV.MAKEKEY (CDR LOGINFO]
                   T)
               (printout T T "User " [SETQ LASTSTRING (SETQ LASTNAME (CONCAT (CAR UPNAME)
                                                                            "."
                                                                    (CDR UPNAME]
                     " logged in." T)
               (SETQ GVUSERNAME UPNAME)
               (SETQ GVPASSWORD PASSKEY))
              (T (printout T EC)
                 (SETQ ALWAYSASK T)
                 (GO LP)))
          (RETURN LASTNAME])
```

(**MT.REMOVE.FRIEND**
```
  [LAMBDA NIL                                              (* bvm%: "17-SEP-83 14:18")
    (\MT.CHANGE.GROUP.COMPONENT (FUNCTION GV.REMOVEFRIEND)
            'REMOVE])
```

(**MT.REMOVE.MEMBER**
```
  [LAMBDA NIL                                              (* bvm%: "17-SEP-83 14:18")
    (\MT.CHANGE.GROUP.COMPONENT (FUNCTION GV.REMOVEMEMBER)
            'REMOVE])
```

(**MT.REMOVE.OWNER**
```
  [LAMBDA NIL                                              (* bvm%: "26-Apr-84 10:44")
    (\MT.CHANGE.GROUP.COMPONENT (FUNCTION GV.REMOVEOWNER)
            'REMOVE])
```

(**MT.TYPE.ENTRY**
```
  [LAMBDA NIL                                              (* bvm%: "23-Mar-84 12:07")
    (PROG (GVNAMETYPE RNAME)
          (DECLARE (SPECVARS GVNAMETYPE RNAME)
              (USEDFREE LASTNAME LASTGROUP LASTSTRING))    (* RNAME is used by \MT.RECEIVE.ENTRY)
          (COND
              ((SETQ RNAME (\MT.READRNAME " for R-Name: " LASTSTRING))
               (.ELLIPSIS.)
               (COND
                   ((EQ (GV.READENTRY RNAME NIL (FUNCTION \MT.RECEIVE.ENTRY))
                        'BadRName)
                    (printout T T "Name not found" T)))
               (SETQ LASTSTRING RNAME])
```

(**MT.TYPE.MEMBERS**
```
  [LAMBDA NIL                                              (* bvm%: "22-Mar-84 18:53")
    (PROG (GVNAMETYPE NAME INFO)
          (DECLARE (SPECVARS GVNAMETYPE)
              (USEDFREE LASTNAME LASTGROUP LASTSTRING))
          (COND
              ((SETQ NAME (\MT.READRNAME " of group: " LASTGROUP))
               (.ELLIPSIS.)
               (GV.READMEMBERS NAME NIL (FUNCTION \MT.PRINTRLIST))
               (SELECTC GVNAMETYPE
                   (\NAMETYPE.GROUP
                       (SETQ LASTGROUP NAME))
                   (\NAMETYPE.INDIVIDUAL
                       (printout T T "Can't: " NAME " is an individual")
                       (SETQ LASTNAME NAME))
                   (printout T T "Name not found"))
               (TERPRI T)
               (SETQ LASTSTRING NAME])

)
```

(DEFINEQ

(\MT.MAYBE.PRINT.OVERSTAMPED.RLIST
```
  [LAMBDA (INSTREAM OUTSTREAM HEADING EVENIFNIL)                (* bvm%: "22-Mar-84 18:49")


          (* * Print a component consisting of an RList, a stamp list, a "removal" RList
          (not interesting) and another stamp list)

     (\MT.PRINTSTRINGLIST INSTREAM OUTSTREAM HEADING EVENIFNIL)
     (\SKIPCOMPONENT INSTREAM)
     (\SKIPCOMPONENT INSTREAM)
     (\SKIPCOMPONENT INSTREAM])
```

(\MT.MAYBE.PRINT.STRING
```
  [LAMBDA (INSTREAM OUTSTREAM HEADING)                          (* bvm%: "20-Jul-85 16:56")
    (PROG (STRLEN)
         (COND
            ((AND (NEQ (\WIN INSTREAM)
                        0)
                  (NEQ (SETQ STRLEN (PROGN (\RECEIVESTAMP INSTREAM T)
                                                            (* Skip stamp)
                                           (\WIN INSTREAM)))
                        0))
              (AND HEADING (PRIN1 HEADING OUTSTREAM))
              (RPTQ STRLEN (\OUTCHAR OUTSTREAM (BIN INSTREAM)))
              (COND
                 ((ODDP STRLEN)
                  (BIN INSTREAM)))
              (AND HEADING (TERPRI OUTSTREAM])
```

(\MT.PRINTRLIST
```
  [LAMBDA (INSTREAM)                                            (* bvm%: "22-Mar-84 18:53")


          (* Response fn for operations that read a list of strings. Expects to see a stamp followed by a list of strings.
          Strings are printed to T)

     (\RECEIVESTAMP INSTREAM T)
     (TERPRI T)
     (\MT.PRINTSTRINGLIST INSTREAM (GETSTREAM T 'OUTPUT])
```

(\MT.PRINTSTRINGLIST
```
  [LAMBDA (INSTREAM OUTSTREAM HEADING EVENIFNIL SEPR)           ; Edited 15-Jun-90 18:29 by jds


          (* * Interprets list of components coming on INSTREAM as a list of strings, and prints them to OUTSTREAM, separating
          strings with SEPR)

     (OR SEPR (SETQ SEPR ", "))
     (PROG ((CNT 0)
            (NWORDS (\WIN INSTREAM))
            STRLEN RMAR)
          (COND
             ((EQ NWORDS 0)
              (COND
                 (EVENIFNIL (printout OUTSTREAM HEADING "null" T)))
              (RETURN 0)))
          (COND
             (HEADING (PRIN1 HEADING OUTSTREAM)))
          (SETQ RMAR (LINELENGTH NIL OUTSTREAM))
          [do (add CNT 1)
              (SETQ STRLEN (\WIN INSTREAM))
              (\WIN INSTREAM)                                   (* ignore maxLength)
              (AND (IGREATERP (IPLUS (IPLUS STRLEN 2)
                                     (fetch (STREAM CHARPOSITION) of OUTSTREAM))
                              RMAR)
                   (FRESHLINE OUTSTREAM))
              (FRPTQ STRLEN (\OUTCHAR OUTSTREAM (BIN INSTREAM)))
              (COND
                 ((ODDP STRLEN)                                 (* read padding)
                  (BIN INSTREAM)))
              (SETQ NWORDS (IDIFFERENCE NWORDS (IPLUS (FOLDHI STRLEN BYTESPERWORD)
                                                      2)))
              (COND
                 ((IGREATERP NWORDS 0)
                  (PRIN1 SEPR OUTSTREAM))
                 (T (RETURN]
          (AND HEADING (TERPRI OUTSTREAM))
          (RETURN CNT])
```

(\SKIPCOMPONENT
```
  [LAMBDA (STREAM)                                              (* bvm%: "20-Jul-85 16:55")
                                                                (* Skips over a component, which is a word count followed by
                                                                that many words)

     (to (\WIN STREAM) do (\WIN STREAM])
```

(**MT.SKIPSTRINGLIST**
```
  [LAMBDA (INSTREAM)                                               (* bvm%: "20-Jul-85 16:55")

          (* * Interprets list of components coming on INSTREAM as a list of strings, and counts them without printing them)

    (bind (CNT _ 0)
          (NWORDS _ (\WIN INSTREAM))
          STRLEN while (IGREATERP NWORDS 0) do (add CNT 1)
                                               (SETQ STRLEN (\WIN INSTREAM))
                                               (\WIN INSTREAM)          (* ignore maxLength)
                                               (FRPTQ STRLEN (BIN INSTREAM))
                                               (COND
                                                  ((ODDP STRLEN)   (* read padding)
                                                   (BIN INSTREAM)))
                                               (SETQ NWORDS (IDIFFERENCE NWORDS (IPLUS (FOLDHI STRLEN
                                                                                              BYTESPERWORD)
                                                                                       2)))
          finally (RETURN CNT])
```

(**MT.RECEIVE.ENTRY**
```
  [LAMBDA (INSTREAM)                                               (* bvm%: "20-Jul-85 16:56")
    (DECLARE (USEDFREE LASTNAME LASTGROUP RNAME))

          (* * Called by GV.READENTRY to parse and display some of what Grapevine sends back as "the entire database entry" for
          NAME. The contents are different for groups, individuals, and dead folk)

    (PROG ((OUTSTREAM (GETSTREAM T 'OUTPUT))
            NAMETYPE)
          (\RECEIVESTAMP INSTREAM T)                               (* Skip stamp)
          (\WIN INSTREAM)                                          (* Skip component count)
          (TERPRI OUTSTREAM)
          (COND
             ((NEQ NAMETYPE \NAMETYPE.NOTFOUND)

          (* There is a database entry. First component is the "prefix" %, which contains, among other things, the name's type and its
          "official" name)

               (\WIN INSTREAM)                                     (* Length of this component)
               (\RECEIVESTAMP INSTREAM T)                          (* Skip stamp)
               (SETQ NAMETYPE (\WIN INSTREAM))
               (printout OUTSTREAM (SETQ RNAME (\RECEIVERNAME INSTREAM))
                       " is ")))
          (SELECTC NAMETYPE
             (\NAMETYPE.INDIVIDUAL
                 (printout OUTSTREAM "an individual" T)
                 (\SKIPCOMPONENT INSTREAM)                         (* Skip password)
                 (\MT.MAYBE.PRINT.STRING INSTREAM OUTSTREAM "Connect-site: ")
                 (\MT.MAYBE.PRINT.OVERSTAMPED.RLIST INSTREAM OUTSTREAM "Forwarding: ")
                 (\MT.MAYBE.PRINT.OVERSTAMPED.RLIST INSTREAM OUTSTREAM "Mailbox-sites: ")
                 (SETQ LASTNAME RNAME))
             (\NAMETYPE.GROUP
                 (printout OUTSTREAM "a group" T)
                 (\MT.MAYBE.PRINT.STRING INSTREAM OUTSTREAM "Remark: ")
                 (printout OUTSTREAM "Number of members: " .P2 (\MT.SKIPSTRINGLIST INSTREAM)
                        T)
                 (\SKIPCOMPONENT INSTREAM)                         (* Skip stamp list)
                 (\SKIPCOMPONENT INSTREAM)                         (* Skip DelMembers)
                 (\SKIPCOMPONENT INSTREAM)                         (* Skip stamp list)
                 (\MT.MAYBE.PRINT.OVERSTAMPED.RLIST INSTREAM OUTSTREAM "Owners: " T)
                 (\MT.MAYBE.PRINT.OVERSTAMPED.RLIST INSTREAM OUTSTREAM "Friends: " T)
                 (SETQ LASTGROUP RNAME))
             (\NAMETYPE.DEAD
                 (printout OUTSTREAM "dead" T))
             NIL])
)

(DEFINEQ

## (\\**MT.READRNAME**
  [LAMBDA (PROMPT DEFAULT)                                         (* bvm%: "20-Jul-85 17:58")
    (PROG ((NAME (PROMPTFORWORD PROMPT DEFAULT NIL T NIL NIL RNAMEDELIMITERS)))
          [COND
             ((NULL NAME)
              (printout T " xxx" T)
              (RETURN))
             ((AND (NOT (STRPOS "." NAME))
                   (NOT (STRPOS "*" NAME)))                        (* No registry included and "name" is not a pattern)
              (printout T "." DEFAULTREGISTRY)
              (SETQ NAME (CONCAT NAME "." DEFAULTREGISTRY]
          (RETURN NAME])
```

(**MT.PERMIT.NS**

```
  [LAMBDA NIL                                                    (* Let users type names with spaces etc in them)
    (SETQ RNAMEDELIMITERS (CHARCODE (CR])
)

(RPAQQ \MT.ELLIPSIS " ... ")

(DECLARE%: EVAL@COMPILE DONTCOPY

(RPAQQ GVNAMETYPES ((\NAMETYPE.GROUP 0)
                    (\NAMETYPE.INDIVIDUAL 1)
                    (\NAMETYPE.NOTFOUND 2)
                    (\NAMETYPE.DEAD 3)))

(DECLARE%: EVAL@COMPILE

(RPAQQ \NAMETYPE.GROUP 0)

(RPAQQ \NAMETYPE.INDIVIDUAL 1)

(RPAQQ \NAMETYPE.NOTFOUND 2)

(RPAQQ \NAMETYPE.DEAD 3)

(CONSTANTS (\NAMETYPE.GROUP 0)
       (\NAMETYPE.INDIVIDUAL 1)
       (\NAMETYPE.NOTFOUND 2)
       (\NAMETYPE.DEAD 3))
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS .ELLIPSIS. MACRO (NIL (printout T \MT.ELLIPSIS)))
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \MT.ELLIPSIS)
)
)

(FILESLOAD GRAPEVINE)

(PUTPROPS MAINTAIN COPYRIGHT ("Venue & Xerox Corporation" 1983 1984 1985 1990))
```

FUNCTION INDEX

CONSTANT INDEX

VARIABLE INDEX

MACRO INDEX