File created: 15-Jun-90 14:49:16 **{DSK}<usr>local>lde>lispcore>internal>library>GRAPEVINE.;2**

changes to: (VARS GRAPEVINECOMS)
(FNS \ENQUIRE \PERFORMGVOP FINDREGSERVER GV.KILLSOCKET \GV.WHENCLOSED)

previous date: 21-May-86 10:53:33 {DSK}<usr>local>lde>lispcore>internal>library>GRAPEVINE.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

;;


```
(RPAQQ GRAPEVINECOMS
       [(COMS (* Functions for interrogating the database)
              (FNS GV.AUTHENTICATE GV.CHECKSTAMP GV.EXPAND GV.IDENTIFYCALLER GV.IDENTIFYME GV.ISINLIST
                   GV.ISMEMBERCLOSURE GV.ISMEMBERDIRECT GV.READCONNECT GV.READENTRY GV.READFRIENDS
                   GV.READMEMBERS GV.READOWNERS GV.READREMARK)
              (* Functions which update the database)
              (FNS GV.ADDFORWARD GV.ADDFRIEND GV.ADDLISTOFMEMBERS GV.ADDMAILBOX GV.ADDMEMBER GV.ADDOWNER
                   GV.CHANGECONNECT GV.CHANGEPASSWORD GV.CHANGEREMARK GV.CREATEGROUP GV.CREATEINDIVIDUAL
                   GV.DELETEGROUP GV.DELETEINDIVIDUAL GV.NEWNAME GV.REMOVEFORWARD GV.REMOVEFRIEND
                   GV.REMOVEMAILBOX GV.REMOVEMEMBER GV.REMOVEOWNER))
        (COMS (* Talking to Reg Servers)
              (FNS \GVOP \ENQUIRE \PERFORMGVOP FINDREGSERVER LOCATESOCKETS)
              (ADDVARS (\GVCONNECTIONS))
              (VARS (REGROOT '(GV . GV))
                    (REGROOTNLSNAME "GrapevineRServer")
                    (\REG.IOTIMEOUT 30000))
              (DECLARE%: DOEVAL@COMPILE DONTCOPY (COMS * GVPROTOCOLDEFS)))
        (COMS (* Making server connections)
              (FNS OPENCLOSESTSOCKET \OPENGVCONNECTION GV.KILLSOCKET \GV.WHENCLOSED)
              (DECLARE%: DOEVAL@COMPILE DONTCOPY (RECORDS GVCONNECTION)
                     (CONSTANTS (\DEFAULTPOLLINGSOC 5))
                     (GLOBALVARS \BETWEENPROBEDELAY \CONNECTTIMEOUT))
              (VARS (\BETWEENPROBEDELAY 1000)
                    (\CONNECTTIMEOUT 30000)))
        (COMS (* Checking arguments)
              (FNS \CHECKNAME \CHECKSTRING \NONAMEERR \UNPACKREG)
              (INITVARS (DEFAULTREGISTRY))
              (GLOBALVARS DEFAULTREGISTRY))
        (COMS (* GVKEY)
              (FNS \CHECKKEY GV.MAKEKEY)
              (DECLARE%: DOEVAL@COMPILE DONTCOPY (RECORDS GVKEY)
                     (CONSTANTS \#BYTES.GVKEY)
                     (MACROS KEYP CREATEKEY GETKEYBYTE SETKEYBYTE))
              (INITRECORDS GVKEY))
        [COMS (* TIMESTAMP)
              (DECLARE%: DOEVAL@COMPILE DONTCOPY (RECORDS TIMESTAMP)
                     (CONSTANTS \#BYTES.TIMESTAMP))
              (INITRECORDS TIMESTAMP)
              (FNS \TIMESTAMP.DEFPRINT \CHECKSTAMP)
              (DECLARE%: DONTEVAL@LOAD DOCOPY (P (DEFPRINT 'TIMESTAMP '\TIMESTAMP.DEFPRINT]
        (COMS (* I/O primitives)
              (FNS \SENDITEM \SENDSTRING)
              (FNS \RECEIVEBOOL \RECEIVECLIST \RECEIVECOMPONENT \RECEIVERLIST \RECEIVERNAME \RECEIVESTAMP
                   \RECEIVESTRING)
              (VARS (\3BYTEKLUDGEKEY '$$3byte$$))
              (DECLARE%: DOEVAL@COMPILE DONTCOPY (MACROS \RECEIVEWORD \SKIPWORD \SENDWORD)
                     (CONSTANTS (\MAXGVSTRING 64))
                     (GLOBALVARS \3BYTEKLUDGEKEY)))
        (DECLARE%: DONTEVAL@LOAD EVAL@COMPILE DONTCOPY (P (SELECTQ (COMPILEMODE)
                                                                  (D (FILESLOAD (LOADCOMP)
                                                                                PUP BSP))
                                                                  (PDP-10 (FILESLOAD (LOADCOMP)
                                                                                     PUP10 BSPAUX))
                                                                  NIL])
```

(* * Functions for interrogating the database)

```
(DEFINEQ
```

(**GV.AUTHENTICATE**

```
  [LAMBDA (NAME KEY)                                        (* ht%: "14-JAN-82 10:24")
    (\GVOP \OP.AUTHENTICATE (\CHECKNAME NAME)
           (LIST (\CHECKKEY KEY])
```

(**GV.CHECKSTAMP**

```
  [LAMBDA (NAME OLDSTAMP)                                   (* ht%: "22-JAN-82 10:07")
    (\GVOP \OP.CHECKSTAMP (\CHECKNAME NAME)
           (LIST (\CHECKSTAMP OLDSTAMP))
```

```
          (FUNCTION \RECEIVESTAMP])
```

(GV.EXPAND

```
  [LAMBDA (NAME OLDSTAMP)                                    (* M.Yonke "10-AUG-83 11:10")

          (* Does the database Expand operation -
          named to avoid conflict with the mail server version (MSExpand))

    (\GVOP \OP.GVEXPAND (\CHECKNAME NAME)
           (LIST (\CHECKSTAMP OLDSTAMP))
           (FUNCTION \RECEIVERLIST])
```

(GV.IDENTIFYCALLER

```
  [LAMBDA (NAME KEY)                                         (* ht%: "14-JAN-82 10:27")
    (\GVOP \OP.IDENTIFYCALLER (\CHECKNAME NAME)
           (LIST (\CHECKKEY KEY])
```

(GV.IDENTIFYME

```
  [LAMBDA NIL                                                (* bvm%: "17-SEP-83 14:14")
                                                             (* Calls GV.IDENTIFYCALLER with info provided by LOGIN)
    (PROG ((npw (\INTERNAL/GETPASSWORD NIL)))
          (RETURN (GV.IDENTIFYCALLER (CAR npw)
                         (CDR npw])
```

(GV.ISINLIST

```
  [LAMBDA (GROUP MEMBER WHAT WHICH WHERE)                    (* bvm%: "21-May-86 10:34")
    (\GVOP \OP.ISINLIST (\CHECKNAME GROUP)
           (LIST (\CHECKSTRING MEMBER)
                 (LIST \3BYTEKLUDGEKEY (OR WHAT OP.ITSELF)
                       (OR WHICH OP.MEMBERS)
                       (OR WHERE OP.DIRECT)))
           (FUNCTION \RECEIVEBOOL])
```

(GV.ISMEMBERCLOSURE

```
  [LAMBDA (GROUP MEMBER)                                     (* bvm%: "21-May-86 10:34")
    (\GVOP \OP.ISMEMBERCLOSURE (\CHECKNAME GROUP)
           (LIST (\CHECKSTRING MEMBER))
           (FUNCTION \RECEIVEBOOL])
```

(GV.ISMEMBERDIRECT

```
  [LAMBDA (GROUP MEMBER)                                     (* bvm%: "21-May-86 10:34")
    (\GVOP \OP.ISMEMBERDIRECT (\CHECKNAME GROUP)
           (LIST (\CHECKSTRING MEMBER))
           (FUNCTION \RECEIVEBOOL])
```

(GV.READCONNECT

```
  [LAMBDA (NAME)                                             (* ht%: "14-JAN-82 10:20")
    (\GVOP \OP.READCONNECT (\CHECKNAME NAME)
           NIL
           (FUNCTION \RECEIVERNAME])
```

(GV.READENTRY

```
  [LAMBDA (NAME OLDSTAMP READFN)                             (* bvm%: "22-Mar-84 14:05")
    (\GVOP \OP.READENTRY (\CHECKNAME NAME)
           (LIST (\CHECKSTAMP OLDSTAMP))
           (OR READFN (FUNCTION \RECEIVECLIST])
```

(GV.READFRIENDS

```
  [LAMBDA (NAME OLDSTAMP READFN)                             (* bvm%: "22-Mar-84 14:03")
    (\GVOP \OP.READFRIENDS (\CHECKNAME NAME)
           (LIST (\CHECKSTAMP OLDSTAMP))
           (OR READFN (FUNCTION \RECEIVERLIST])
```

(GV.READMEMBERS

```
  [LAMBDA (NAME OLDSTAMP READFN)                             (* bvm%: "22-Mar-84 14:03")
    (\GVOP \OP.READMEMBERS (\CHECKNAME NAME)
           (LIST (\CHECKSTAMP OLDSTAMP))
           (OR READFN (FUNCTION \RECEIVERLIST])
```

(GV.READOWNERS

```
  [LAMBDA (NAME OLDSTAMP READFN)                             (* bvm%: "22-Mar-84 14:04")
    (\GVOP \OP.READOWNERS (\CHECKNAME NAME)
           (LIST (\CHECKSTAMP OLDSTAMP))
           (OR READFN (FUNCTION \RECEIVERLIST])
```

```
(GV.READREMARK
  [LAMBDA (NAME)                                                    (* ht%: "14-JAN-82 10:21")
    (\GVOP \OP.READREMARK (\CHECKNAME NAME)
           NIL
           (FUNCTION \RECEIVERNAME])

)


          (* * Functions which update the database)

(DEFINEQ

(GV.ADDFORWARD
  [LAMBDA (NAME STRING IDENTIFYUSER PASSWORD)                       (* bvm%: "16-SEP-83 18:28")
    (\GVOP \OP.ADDFORWARD (\CHECKNAME NAME)
           (LIST (\CHECKSTRING STRING))
           NIL
           (OR IDENTIFYUSER T)
           PASSWORD])

(GV.ADDFRIEND
  [LAMBDA (GROUP FRIEND IDENTIFYUSER PASSWORD)                      (* bvm%: "21-May-86 10:38")
    (\GVOP \OP.ADDFRIEND (\CHECKNAME GROUP)
           (LIST (\CHECKSTRING FRIEND))
           NIL
           (OR IDENTIFYUSER T)
           PASSWORD])

(GV.ADDLISTOFMEMBERS
  [LAMBDA (GROUP MEMBERS IDENTIFYUSER PASSWORD)                     (* bvm%: "21-May-86 10:39")
    (\GVOP \OP.ADDLISTOFMEMBERS (\CHECKNAME GROUP)
           [LIST (COND
                    ([AND (LISTP MEMBERS)
                          (OR (STRINGP (CAR MEMBERS))
                              (LITATOM (CAR MEMBERS)))
                          (for p on MEMBERS when (CDR p) always (AND (OR (STRINGP (CADR p))
                                                                         (LITATOM (CADR p)))
                                                                     (ALPHORDER (CAR p)
                                                                            (CADR p]
                     MEMBERS)
                    (T (ERROR "must have ordered list of strings" MEMBERS]
           NIL
           (OR IDENTIFYUSER T)
           PASSWORD])

(GV.ADDMAILBOX
  [LAMBDA (NAME STRING IDENTIFYUSER PASSWORD)                       (* bvm%: "16-SEP-83 18:20")
    (\GVOP \OP.ADDMAILBOX (\CHECKNAME NAME)
           (LIST (\CHECKSTRING STRING))
           NIL
           (OR IDENTIFYUSER T)
           PASSWORD])

(GV.ADDMEMBER
  [LAMBDA (GROUP MEMBER IDENTIFYUSER PASSWORD)                      (* bvm%: "21-May-86 10:39")
    (\GVOP \OP.ADDMEMBER (\CHECKNAME GROUP)
           (LIST (\CHECKSTRING MEMBER))
           NIL
           (OR IDENTIFYUSER T)
           PASSWORD])

(GV.ADDOWNER
  [LAMBDA (GROUP OWNER IDENTIFYUSER PASSWORD)                       (* bvm%: "21-May-86 10:39")
    (\GVOP \OP.ADDOWNER (\CHECKNAME GROUP)
           (LIST (\CHECKSTRING OWNER))
           NIL
           (OR IDENTIFYUSER T)
           PASSWORD])

(GV.CHANGECONNECT
  [LAMBDA (NAME SITE IDENTIFYUSER PASSWORD)                         (* bvm%: "16-SEP-83 18:27")
    (\GVOP \OP.CHANGECONNECT (\CHECKNAME NAME)
           [LIST (OR (STRINGP SITE)
                     (AND (LITATOM SITE)
                          SITE)
                     (COND
                        ((AND [OR (LISTP SITE)
                                  (NUMBERP SITE)
                                  (AND (NOT SITE)
                                       (SETQ SITE (\LOCALPUPADDRESS]
```

```
                            (PORTSTRING SITE)))
                       (T (ERROR "Invalid Site" SITE]
          NIL IDENTIFYUSER PASSWORD])
```

(GV.CHANGEPASSWORD

```
  [LAMBDA (NAME KEY IDENTIFYUSER PASSWORD)                    (* bvm%: "16-SEP-83 18:21")
    (\GVOP \OP.CHANGEPASSWORD (\CHECKNAME NAME)
           (LIST (\CHECKKEY KEY))
           NIL
           (OR IDENTIFYUSER T)
           PASSWORD])
```

(GV.CHANGEREMARK

```
  [LAMBDA (NAME STRING IDENTIFYUSER PASSWORD)                 (* bvm%: "16-SEP-83 18:22")
    (\GVOP \OP.CHANGEREMARK (\CHECKNAME NAME)
           (LIST (\CHECKSTRING STRING))
           NIL
           (OR IDENTIFYUSER T)
           PASSWORD])
```

(GV.CREATEGROUP

```
  [LAMBDA (NAME IDENTIFYUSER PASSWORD)                        (* bvm%: "16-SEP-83 18:22")
    (\GVOP \OP.CREATEGROUP (\CHECKNAME NAME)
           NIL NIL (OR IDENTIFYUSER T)
           PASSWORD])
```

(GV.CREATEINDIVIDUAL

```
  [LAMBDA (NAME KEY IDENTIFYUSER PASSWORD)                    (* bvm%: "16-SEP-83 18:23")
    (\GVOP \OP.CREATEINDIVIDUAL (\CHECKNAME NAME)
           (LIST (\CHECKKEY KEY))
           NIL
           (OR IDENTIFYUSER T)
           PASSWORD])
```

(GV.DELETEGROUP

```
  [LAMBDA (NAME IDENTIFYUSER PASSWORD)                        (* bvm%: "16-SEP-83 18:23")
    (\GVOP \OP.DELETEGROUP (\CHECKNAME NAME)
           NIL NIL (OR IDENTIFYUSER T)
           PASSWORD])
```

(GV.DELETEINDIVIDUAL

```
  [LAMBDA (NAME IDENTIFYUSER PASSWORD)                        (* bvm%: "16-SEP-83 18:23")
    (\GVOP \OP.DELETEINDIVIDUAL (\CHECKNAME NAME)
           NIL NIL (OR IDENTIFYUSER T)
           PASSWORD])
```

(GV.NEWNAME

```
  [LAMBDA (NAME GV.NEWNAME IDENTIFYUSER PASSWORD)             (* bvm%: "16-SEP-83 18:24")
    (\GVOP \OP.NEWNAME (\CHECKNAME NAME)
           (LIST (\CHECKNAME GV.NEWNAME))
           NIL
           (OR IDENTIFYUSER T)
           PASSWORD])
```

(GV.REMOVEFORWARD

```
  [LAMBDA (NAME STRING IDENTIFYUSER PASSWORD)                 (* bvm%: "16-SEP-83 18:24")
    (\GVOP \OP.REMOVEFORWARD (\CHECKNAME NAME)
           (LIST (\CHECKSTRING STRING))
           NIL
           (OR IDENTIFYUSER T)
           PASSWORD])
```

(GV.REMOVEFRIEND

```
  [LAMBDA (GROUP FRIEND IDENTIFYUSER PASSWORD)                (* bvm%: "21-May-86 10:40")
    (\GVOP \OP.REMOVEFRIEND (\CHECKNAME GROUP)
           (LIST (\CHECKSTRING FRIEND))
           NIL
           (OR IDENTIFYUSER T)
           PASSWORD])
```

(GV.REMOVEMAILBOX

```
  [LAMBDA (NAME STRING IDENTIFYUSER PASSWORD)                 (* bvm%: "16-SEP-83 18:25")
    (\GVOP \OP.REMOVEMAILBOX (\CHECKNAME NAME)
           (LIST (\CHECKSTRING STRING))
           NIL
           (OR IDENTIFYUSER T)
           PASSWORD])
```

```
(GV.REMOVEMEMBER
  [LAMBDA (GROUP MEMBER IDENTIFYUSER PASSWORD)                    (* bvm%: "21-May-86 10:40")
    (\GVOP \OP.REMOVEMEMBER (\CHECKNAME GROUP)
           (LIST (\CHECKSTRING MEMBER))
           NIL
           (OR IDENTIFYUSER T)
           PASSWORD])

(GV.REMOVEOWNER
  [LAMBDA (GROUP OWNER IDENTIFYUSER PASSWORD)                     (* bvm%: "21-May-86 10:40")
    (\GVOP \OP.REMOVEOWNER (\CHECKNAME GROUP)
           (LIST (\CHECKSTRING OWNER))
           NIL
           (OR IDENTIFYUSER T)
           PASSWORD])

)


          (* * Talking to Reg Servers)

(DEFINEQ

(\GVOP
  [LAMBDA (OP name itemList READFN IDENTIFYUSER PASSWORD)         (* bvm%: "22-Mar-84 14:55")

          (* Supervises a registration database operation. Does the initial interaction, applies READFN to the input side of the
          connection to collect results, and interprets same if necessary)

    (\ENQUIRE name (CONS OP (CONS name itemList))
           READFN IDENTIFYUSER PASSWORD])

(\ENQUIRE
  [LAMBDA (NAME ARGS READFN IDENTIFYUSER PASSWORD)                ; Edited 15-Jun-90 14:27 by jds

          (* Attempt to accomplish some interaction with a reg. server. Implements the Taft/Birrell approach of first trying anybody
          we're connected to, failing that trying the closest reg. server we can find, and only if that fails as well do we get down to
          basics and actually go thru the lookup procedure to find someone who knows what we need)

    (PROG ((REGISTRY REGROOT)
           RESULT CONN INFO)
      LP   (COND
              ((NOT (SETQ CONN (FINDREGSERVER REGISTRY)))
               (RETURN EC.ALLDOWN)))
           [COND
              (IDENTIFYUSER [COND
                               ((EQ IDENTIFYUSER T)
                                (SETQ INFO (\INTERNAL/GETPASSWORD))
                                (SETQ IDENTIFYUSER (CAR INFO]
                     (COND
                        ((AND (NEQ (fetch (GVCONNECTION GVIDENTIFIED) of CONN)
                                   IDENTIFYUSER)
                              (NOT (EQUAL (fetch (GVCONNECTION GVIDENTIFIED) of CONN)
                                          IDENTIFYUSER)))
                         (COND
                            ([NOT (SETQ RESULT (\PERFORMGVOP CONN (LIST \OP.IDENTIFYCALLER (\CHECKNAME
                                                                                                IDENTIFYUSER)
                                                                                           (\CHECKKEY (OR PASSWORD (CDR INFO]
                             (BLOCK)
                             (replace (GVCONNECTION GVBUSY) of CONN with NIL)
                             (GO LP))
                            ((SETQ RESULT (SELECTC (fetch HIBYTE of RESULT)
                                                   (\RC.BADRNAME EC.BADRNAME)
                                                   (\RC.BADPASSWORD
                                                       EC.BADPASSWORD)
                                                   (\RC.ALLDOWN EC.ALLDOWN)
                                                   (\RC.DONE NIL)
                                                   (SHOULDNT)))
                             (RETURN RESULT))
                            (T (replace (GVCONNECTION GVIDENTIFIED) of CONN with IDENTIFYUSER]
           (SETQ RESULT (SELECTC (COND
                                    ((SETQ RESULT (\PERFORMGVOP CONN ARGS))
                                                                  (* we ignore the name type and return the code part of the
                                                                  return code)
                                     (SETQ GVNAMETYPE (fetch LOBYTE of RESULT))
                                     (SETQ RESULT (fetch HIBYTE of RESULT)))
                                    (T

          (* The usual causes for this are the stream is not in fact open despite our efforts to insure that it is, or that the other end has
          gone to sleep and the BSPIOTIMEOUT occurs. If this happens too often, \REG.IOTIMEOUT should be lengthened)

                                       (BLOCK)                    (* Let RTP run and clean this guy out)
                                       (replace (GVCONNECTION GVBUSY) of CONN with NIL)
```

```
                                (GO LP)))
                           (\RC.NOCHANGE                    (* For use with timestamps, says entry has not changed, so no
                                                               values to return)
                                         EC.NOCHANGE)
                           (\RC.DONE (COND
                                        (READFN (APPLY* READFN (fetch (GVCONNECTION GVINSTREAM) of CONN)))
                                        (T T)))
                           (\RC.WRONGSERVER                 (* so we have to do it right after all)
                                (COND
                                   ((NEQ REGISTRY REGROOT)
                                    EC.BADRNAME)
                                   (T (replace (GVCONNECTION GVBUSY) of CONN with NIL)
                                      (SETQ REGISTRY (CONS (CDR NAME)
                                                           'GV))
                                      (GO LP))))
                           (\RC.BADRNAME EC.BADRNAME)
                           (\RC.NOTALLOWED
                                EC.NOTALLOWED)
                           (\RC.BADPASSWORD
                                EC.BADPASSWORD)
                           (\RC.ALLDOWN EC.ALLDOWN)
                           RESULT))
        (replace (GVCONNECTION GVBUSY) of CONN with NIL)
        (RETURN RESULT])
```

(\PERFORMGVOP

```
  [LAMBDA (CONN ARGS)                                       ; Edited 15-Jun-90 14:27 by jds
    (CAR (NLSETQ (LET ((STREAM (fetch (GVCONNECTION GVOUTSTREAM) of CONN)))
                      (for e in ARGS do (\SENDITEM STREAM e))
                      (FORCEOUTPUT STREAM)
                      (\RECEIVEWORD (fetch (GVCONNECTION GVINSTREAM) of CONN])
```

(FINDREGSERVER

```
  [LAMBDA (REGISTRY ERRORFLG)                               ; Edited 15-Jun-90 14:27 by jds
                                                            (* Find a registration server for REGISTRY -
                                                            the closest one available)
    (PROG (NEWSOC)
          [COND
             ((NLISTP REGISTRY)
              (SETQ REGISTRY (\UNPACKREG REGISTRY]
          (RETURN (COND
                     [(UNINTERRUPTABLY
                          (for CONN in \GVCONNECTIONS when [AND (NULL (fetch (GVCONNECTION GVBUSY) of CONN))
                                                                (OR (EQ REGISTRY REGROOT)
                                                                    (EQUAL REGISTRY (fetch (GVCONNECTION
                                                                                              GVREGISTRY)
                                                                                       of CONN]
                             do (replace (GVCONNECTION GVBUSY) of CONN with T)
                                (RETURN CONN)))]
                     ((SETQ NEWSOC (OPENCLOSESTSOCKET (LOCATESOCKETS REGISTRY ERRORFLG)
                                       \REG.SERVERPOLLINGSOC \REG.SERVERENQUIRYSOC \REG.IOTIMEOUT))
                      (replace (GVCONNECTION GVREGISTRY) of NEWSOC with REGISTRY)
                      (replace (GVCONNECTION GVBUSY) of NEWSOC with T)
                      (push \GVCONNECTIONS NEWSOC)
                      NEWSOC)
                     (ERRORFLG (ERROR "Couldn't open connection for" REGISTRY])
```

(LOCATESOCKETS

```
  [LAMBDA (SITE ERRORFLG)                                   (* bvm%: "17-SEP-83 14:15")

          (* get a list of sockets for a SITE -
          a three step process (except for GV.GV) -
          find the members of the site, find the connect sites for each, turn those into sockets)

    (COND
       ((EQUAL SITE REGROOT)                                (* treat the root -
                                                            "GV.GV" -
                                                            specially)
        (ETHERPORT REGROOTNLSNAME ERRORFLG T))
       (T (bind cn for rName in [CDR (OR (LISTP (GV.READMEMBERS SITE))
                                         (COND
                                            (ERRORFLG (ERROR "Not a valid site" SITE]
             join (OR (AND (SETQ cn (STRINGP (GV.READCONNECT rName)))
                           (ETHERPORT cn NIL T))
                      (ETHERPORT rName NIL T)
                      (COND
                         (ERRORFLG (HELP "Can't look up connect name" (CONS rName cn])

)

(ADDTOVAR \GVCONNECTIONS )

(RPAQQ REGROOT (GV . GV))
```

```
(RPAQ REGROOTNLSNAME "GrapevineRServer")

(RPAQQ \REG.IOTIMEOUT 30000)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(RPAQQ GVPROTOCOLDEFS
       ((CONSTANTS * \GV.OPS)
        (* Grapevine response codes)
        (CONSTANTS * \GV.RESPONSES)
        (* Response codes the user sees)
        (CONSTANTS * \GVU.RESPONSES)
        (GLOBALVARS REGROOT REGROOTNLSNAME \REG.IOTIMEOUT \GVCONNECTIONS)
        (CONSTANTS (\REG.SERVERENQUIRYSOC 40)
               (\REG.SERVERPOLLINGSOC 42))
        (* Constants for calling GV.ISINLIST)
        (CONSTANTS * \GVU.MEMBEROPS)))

(RPAQQ \GV.OPS
       ((\OP.GVEXPAND 1)
        (\OP.READMEMBERS 2)
        (\OP.READOWNERS 3)
        (\OP.READFRIENDS 4)
        (\OP.READENTRY 5)
        (\OP.CHECKSTAMP 6)
        (\OP.READCONNECT 7)
        (\OP.READREMARK 8)
        (\OP.AUTHENTICATE 9)
        (\OP.IDENTIFYCALLER 33)
        (\OP.ISMEMBERDIRECT 40)
        (\OP.ISOWNERDIRECT 41)
        (\OP.ISFRIENDDIRECT 42)
        (\OP.ISMEMBERCLOSURE 43)
        (\OP.ISOWNERCLOSURE 44)
        (\OP.ISFRIENDCLOSURE 45)
        (\OP.ISINLIST 46)
        (\OP.CREATEINDIVIDUAL 12)
        (\OP.DELETEINDIVIDUAL 13)
        (\OP.CREATEGROUP 14)
        (\OP.DELETEGROUP 15)
        (\OP.CHANGEPASSWORD 16)
        (\OP.CHANGECONNECT 17)
        (\OP.CHANGEREMARK 18)
        (\OP.ADDMEMBER 19)
        (\OP.ADDMAILBOX 20)
        (\OP.ADDFORWARD 21)
        (\OP.ADDOWNER 22)
        (\OP.ADDFRIEND 23)
        (\OP.REMOVEMEMBER 24)
        (\OP.REMOVEMAILBOX 25)
        (\OP.REMOVEFORWARD 26)
        (\OP.REMOVEOWNER 27)
        (\OP.REMOVEFRIEND 28)
        (\OP.ADDSELF 29)
        (\OP.REMOVESELF 30)
        (\OP.ADDLISTOFMEMBERS 31)
        (\OP.NEWNAME 32)))

(DECLARE%: EVAL@COMPILE

(RPAQQ \OP.GVEXPAND 1)

(RPAQQ \OP.READMEMBERS 2)

(RPAQQ \OP.READOWNERS 3)

(RPAQQ \OP.READFRIENDS 4)

(RPAQQ \OP.READENTRY 5)

(RPAQQ \OP.CHECKSTAMP 6)

(RPAQQ \OP.READCONNECT 7)

(RPAQQ \OP.READREMARK 8)

(RPAQQ \OP.AUTHENTICATE 9)

(RPAQQ \OP.IDENTIFYCALLER 33)

(RPAQQ \OP.ISMEMBERDIRECT 40)

(RPAQQ \OP.ISOWNERDIRECT 41)

(RPAQQ \OP.ISFRIENDDIRECT 42)

(RPAQQ \OP.ISMEMBERCLOSURE 43)
```

(RPAQQ **OP.ISOWNERCLOSURE** 44)

(RPAQQ **OP.ISFRIENDCLOSURE** 45)

(RPAQQ **OP.ISINLIST** 46)

(RPAQQ **OP.CREATEINDIVIDUAL** 12)

(RPAQQ **OP.DELETEINDIVIDUAL** 13)

(RPAQQ **OP.CREATEGROUP** 14)

(RPAQQ **OP.DELETEGROUP** 15)

(RPAQQ **OP.CHANGEPASSWORD** 16)

(RPAQQ **OP.CHANGECONNECT** 17)

(RPAQQ **OP.CHANGEREMARK** 18)

(RPAQQ **OP.ADDMEMBER** 19)

(RPAQQ **OP.ADDMAILBOX** 20)

(RPAQQ **OP.ADDFORWARD** 21)

(RPAQQ **OP.ADDOWNER** 22)

(RPAQQ **OP.ADDFRIEND** 23)

(RPAQQ **OP.REMOVEMEMBER** 24)

(RPAQQ **OP.REMOVEMAILBOX** 25)

(RPAQQ **OP.REMOVEFORWARD** 26)

(RPAQQ **OP.REMOVEOWNER** 27)

(RPAQQ **OP.REMOVEFRIEND** 28)

(RPAQQ **OP.ADDSELF** 29)

(RPAQQ **OP.REMOVESELF** 30)

(RPAQQ **OP.ADDLISTOFMEMBERS** 31)

(RPAQQ **OP.NEWNAME** 32)

```
(CONSTANTS (\OP.GVEXPAND 1)
       (\OP.READMEMBERS 2)
       (\OP.READOWNERS 3)
       (\OP.READFRIENDS 4)
       (\OP.READENTRY 5)
       (\OP.CHECKSTAMP 6)
       (\OP.READCONNECT 7)
       (\OP.READREMARK 8)
       (\OP.AUTHENTICATE 9)
       (\OP.IDENTIFYCALLER 33)
       (\OP.ISMEMBERDIRECT 40)
       (\OP.ISOWNERDIRECT 41)
       (\OP.ISFRIENDDIRECT 42)
       (\OP.ISMEMBERCLOSURE 43)
       (\OP.ISOWNERCLOSURE 44)
       (\OP.ISFRIENDCLOSURE 45)
       (\OP.ISINLIST 46)
       (\OP.CREATEINDIVIDUAL 12)
       (\OP.DELETEINDIVIDUAL 13)
       (\OP.CREATEGROUP 14)
       (\OP.DELETEGROUP 15)
       (\OP.CHANGEPASSWORD 16)
       (\OP.CHANGECONNECT 17)
       (\OP.CHANGEREMARK 18)
       (\OP.ADDMEMBER 19)
       (\OP.ADDMAILBOX 20)
       (\OP.ADDFORWARD 21)
       (\OP.ADDOWNER 22)
       (\OP.ADDFRIEND 23)
       (\OP.REMOVEMEMBER 24)
       (\OP.REMOVEMAILBOX 25)
       (\OP.REMOVEFORWARD 26)
       (\OP.REMOVEOWNER 27)
       (\OP.REMOVEFRIEND 28)
       (\OP.ADDSELF 29)
       (\OP.REMOVESELF 30)
       (\OP.ADDLISTOFMEMBERS 31)
       (\OP.NEWNAME 32))
```

```
)

          (* * Grapevine response codes)

(RPAQQ \GV.RESPONSES
       ((\RC.DONE 0)
        (\RC.NOCHANGE 1)
        (\RC.OUTOFDATE 2)
        (\RC.NOTALLOWED 3)
        (\RC.BADOPERATION 4)
        (\RC.BADPROTOCOL 5)
        (\RC.BADRNAME 6)
        (\RC.BADPASSWORD 7)
        (\RC.WRONGSERVER 8)
        (\RC.ALLDOWN 9)))

(DECLARE%: EVAL@COMPILE

(RPAQQ \RC.DONE 0)

(RPAQQ \RC.NOCHANGE 1)

(RPAQQ \RC.OUTOFDATE 2)

(RPAQQ \RC.NOTALLOWED 3)

(RPAQQ \RC.BADOPERATION 4)

(RPAQQ \RC.BADPROTOCOL 5)

(RPAQQ \RC.BADRNAME 6)

(RPAQQ \RC.BADPASSWORD 7)

(RPAQQ \RC.WRONGSERVER 8)

(RPAQQ \RC.ALLDOWN 9)

(CONSTANTS (\RC.DONE 0)
       (\RC.NOCHANGE 1)
       (\RC.OUTOFDATE 2)
       (\RC.NOTALLOWED 3)
       (\RC.BADOPERATION 4)
       (\RC.BADPROTOCOL 5)
       (\RC.BADRNAME 6)
       (\RC.BADPASSWORD 7)
       (\RC.WRONGSERVER 8)
       (\RC.ALLDOWN 9))
)

          (* * Response codes the user sees)

(RPAQQ \GVU.RESPONSES
       ((EC.STREAMLOST 'StreamLost)
        (EC.ALLDOWN 'AllDown)
        (EC.NOCHANGE 'NoChange)
        (EC.BADRNAME 'BadRName)
        (EC.BADPASSWORD 'BadPassword)
        (EC.NOTALLOWED 'NotAllowed)))

(DECLARE%: EVAL@COMPILE

(RPAQQ EC.STREAMLOST StreamLost)

(RPAQQ EC.ALLDOWN AllDown)

(RPAQQ EC.NOCHANGE NoChange)

(RPAQQ EC.BADRNAME BadRName)

(RPAQQ EC.BADPASSWORD BadPassword)

(RPAQQ EC.NOTALLOWED NotAllowed)

(CONSTANTS (EC.STREAMLOST 'StreamLost)
       (EC.ALLDOWN 'AllDown)
       (EC.NOCHANGE 'NoChange)
       (EC.BADRNAME 'BadRName)
       (EC.BADPASSWORD 'BadPassword)
       (EC.NOTALLOWED 'NotAllowed))
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS REGROOT REGROOTNLSNAME \REG.IOTIMEOUT \GVCONNECTIONS)
```

```
)

(DECLARE%: EVAL@COMPILE

(RPAQQ \REG.SERVERENQUIRYSOC 40)

(RPAQQ \REG.SERVERPOLLINGSOC 42)

(CONSTANTS (\REG.SERVERENQUIRYSOC 40)
       (\REG.SERVERPOLLINGSOC 42))
)
```

(* * Constants for calling GV.ISINLIST)

```
(RPAQQ \GVU.MEMBEROPS ((OP.ITSELF 0)
                       (OP.ITSREGISTRY 1)
                       (OP.MEMBERS 0)
                       (OP.OWNERS 1)
                       (OP.FRIENDS 2)
                       (OP.DIRECT 0)
                       (OP.CLOSURE 1)
                       (OP.UPARROW 2)))

(DECLARE%: EVAL@COMPILE

(RPAQQ OP.ITSELF 0)

(RPAQQ OP.ITSREGISTRY 1)

(RPAQQ OP.MEMBERS 0)

(RPAQQ OP.OWNERS 1)

(RPAQQ OP.FRIENDS 2)

(RPAQQ OP.DIRECT 0)

(RPAQQ OP.CLOSURE 1)

(RPAQQ OP.UPARROW 2)

(CONSTANTS (OP.ITSELF 0)
       (OP.ITSREGISTRY 1)
       (OP.MEMBERS 0)
       (OP.OWNERS 1)
       (OP.FRIENDS 2)
       (OP.DIRECT 0)
       (OP.CLOSURE 1)
       (OP.UPARROW 2))
)
)
```

(* * Making server connections)

```
(DEFINEQ
```

(OPENCLOSESTSOCKET

```
  [LAMBDA (PORTLIST POLLSOC CONNSOC TIMEOUT)                    (* bvm%: "19-Jul-85 12:42")
```

(* Open a BSP connection with the "closest" respondant on portList.
EchoMe polling to determine responsiveness is to pollSoc, connection will go to connSoc.
We poll in order from nearest to farest by hop order, use broadcast on local net if appropriate, and hope not to engage too many folks before the real thing comes along. The basic structure of this is owed to Taft)

```
    (RESETLST
        [PROG ((MYNET (\LOCALPUPNETNUMBER))
               (BETWEENPROBE (SETUPTIMER 0))
               (PROBECOUNT 1)
               LOCALPORTS ALLPORTS SOC CNTIME REMAININGPORTS PORT VAL PUP)
              [for PORT in PORTLIST do (COND
                                          ((AND POLLSOC (EQ (fetch PUPNET# of (CAR PORT))
                                                            MYNET))
                                           (push LOCALPORTS PORT))
                                          (T (push ALLPORTS PORT]
              [COND
                 (ALLPORTS (SETQ ALLPORTS (SORT.PUPHOSTS.BY.DISTANCE ALLPORTS]
              (COND
                 [LOCALPORTS
```

(* if there is more than one local host on the list, remove them and add a broadcast port for cheaper poll)

```
                          (SETQ ALLPORTS (COND
                                            ((CDR LOCALPORTS)
                                             (CONS (LIST (create PUPADDRESS
                                                                 PUPNET# _ MYNET
```

```
                                                        PUPHOST# _ 0))
                                              ALLPORTS))
                                        (T (APPEND LOCALPORTS ALLPORTS]
                ((NULL ALLPORTS)
                 (RETURN)))
            [RESETSAVE NIL (LIST 'CLOSEPUPSOCKET (SETQ SOC (OPENPUPSOCKET]
            (SETQ CNTIME (SETUPTIMER \CONNECTTIMEOUT))
            (SETQ REMAININGPORTS ALLPORTS)
            (RETURN (do [COND
                           ((TIMEREXPIRED? BETWEENPROBE)
                            [COND
                               ((EQ (SETQ PROBECOUNT (SUB1 PROBECOUNT))
                                    0)
                                (SETQ PORT (CAR REMAININGPORTS))
                                (SETQ PROBECOUNT (COND
                                                    ((EQ (fetch PUPNET# of (CAR PORT))
                                                         MYNET)
                                                     1)
                                                    (T          (* Try twice for hosts not on local net)
                                                       2)))
                                (SETQ REMAININGPORTS (OR (CDR REMAININGPORTS)
                                                         ALLPORTS]
                               (SETQ PUP (ALLOCATE.PUP))
                               (SETUPPUP PUP (CAR PORT)
                                      (OR POLLSOC (CDR PORT)
                                          \DEFAULTPOLLINGSOC)
                                      \PT.ECHOME NIL SOC 'FREE)
                               (SENDPUP SOC PUP)
                               (SETQ BETWEENPROBE (SETUPTIMER \BETWEENPROBEDELAY BETWEENPROBE]
                         (BLOCK)
                         (COND
                            ((AND (SETQ PUP (GETPUP SOC))
                                  (EQ (fetch PUPTYPE of PUP)
                                      \PT.IAMECHO)
                                  (OR (NEQ (fetch PUPSOURCENET of PUP)
                                           MYNET)
                                      (ASSOC (fetch PUPSOURCE of PUP)
                                             LOCALPORTS))
                                  (SETQ VAL (\OPENGVCONNECTION (CONS (fetch PUPSOURCE of PUP)
                                                                     (OR CONNSOC (fetch PUPSOURCESOCKET
                                                                                    of PUP)))
                                                               TIMEOUT)))
```

(* We got back an echo and succeeded in opening a connection.
ASSOC test assures that we don't pay attention to broadcast replies from hosts that we weren't planning to talk to in the first place)

```
                               (RETURN VAL)))
                     repeatuntil (TIMEREXPIRED? CNTIME])])
```

(\OPENGVCONNECTION

```
  [LAMBDA (FRNSOCKET TIMEOUT ERRORHANDLER FAILURESTRING)          (* bvm%: " 4-Feb-86 12:38")
    (LET ((INSTREAM (OPENBSPSTREAM FRNSOCKET NIL ERRORHANDLER TIMEOUT NIL (FUNCTION \GV.WHENCLOSED)
                                   FAILURESTRING)))
         (AND INSTREAM (COND
                          ((STREAMP INSTREAM)
                           (create GVCONNECTION
                                   GVINSTREAM _ INSTREAM
                                   GVOUTSTREAM _ (BSPOUTPUTSTREAM INSTREAM)))
                          (T                                      (* Failed)
                             INSTREAM])
```

(GV.KILLSOCKET

```
  [LAMBDA (SOCKET TIMEOUT)                                        ; Edited 15-Jun-90 14:27 by jds
    (CLOSEBSPSTREAM (fetch (GVCONNECTION GVINSTREAM) of SOCKET)
           TIMEOUT)
    (BLOCK])
```

(\GV.WHENCLOSED

```
  [LAMBDA (BSPSTREAM)                                             ; Edited 15-Jun-90 14:27 by jds
                                                                  (* Called when BSPSTREAM is killed)
    (for CONN in \GVCONNECTIONS when (EQ (fetch (GVCONNECTION GVINSTREAM) of CONN)
                                         BSPSTREAM)
       do (replace (GVCONNECTION GVIDENTIFIED) of CONN with NIL)
          (SETQ \GVCONNECTIONS (DREMOVE CONN \GVCONNECTIONS])

)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(DECLARE%: EVAL@COMPILE

(RECORD GVCONNECTION (GVINSTREAM GVOUTSTREAM GVBUSY GVREGISTRY GVHOPS GVIDENTIFIED))
)
```

```
(DECLARE%: EVAL@COMPILE

(RPAQQ \DEFAULTPOLLINGSOC 5)

(CONSTANTS (\DEFAULTPOLLINGSOC 5))
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \BETWEENPROBEDELAY \CONNECTTIMEOUT)
)
)

(RPAQQ \BETWEENPROBEDELAY 1000)

(RPAQQ \CONNECTTIMEOUT 30000)
```

(* * Checking arguments)

```
(DEFINEQ
```

(\CHECKNAME

```
  [LAMBDA (NAME)                                          (* bvm%: "17-SEP-83 14:37")
    [COND
       ((NLISTP NAME)
        (SETQ NAME (\UNPACKREG (OR NAME (\NONAMEERR]
    (COND
       ((ILESSP (IPLUS (NCHARS (CAR NAME))
                       (NCHARS (CDR NAME)))
                \MAXGVSTRING)                             (* less than because the dot takes 1 more)
        NAME)
       (T (ERROR "name too long - must be < 65 chars" NAME])
```

(\CHECKSTRING

```
  [LAMBDA (STRING)                                        (* Beau " 7-SEP-82 13:43")
    (SELECTQ (TYPENAME STRING)
         (STRINGP)
         (LISTP (COND
                   [(AND (CAR STRING)
                         (LITATOM (CAR STRING))
                         (CDR STRING)
                         (LITATOM (CDR STRING)))
                    (SETQ STRING (CONCAT (CAR STRING)
                                         '%.
                                         (CDR STRING]
                   (T (ERROR "bad string arg" STRING))))
         (LITATOM (SETQ STRING (MKSTRING STRING)))
         (ERROR "bad string arg" STRING))
    (COND
       ((IGREATERP (NCHARS STRING)
                   \MAXGVSTRING)
        (ERROR "string too long" STRING))
       (T STRING])
```

(\NONAMEERR

```
  [LAMBDA NIL                                             (* ht%: "13-JAN-82 12:05")
    (ERROR "must have name for GV user op"])
```

(\UNPACKREG

```
  [LAMBDA (REG)                                           (* bvm%: "20-Jul-85 17:11")
    (LET ((PPOS (STRPOS "." REG)))
         (COND
            [PPOS (CONS (SUBATOM REG 1 (SUB1 PPOS))
                        (SUBATOM REG (ADD1 PPOS]
            (T (CONS (MKATOM REG)
                     DEFAULTREGISTRY])

)

(RPAQ? DEFAULTREGISTRY )

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS DEFAULTREGISTRY)
)
```

(* * GVKEY)

```
(DEFINEQ
```

(\CHECKKEY

```
  [LAMBDA (KEY)                                                    (* bvm%: "17-SEP-83 14:18")
    (COND
       ((KEYP KEY)
        KEY)
       (T (GV.MAKEKEY KEY])
```

(GV.MAKEKEY

```
  [LAMBDA (STRING ISCLEAR)                                         (* bvm%: "19-Jul-85 16:42")
                                                                   (* As per section 2 of the Grapevine Interface document)
    (for I from 0 bind J C (KEY _ (CREATEKEY)) while (SETQ C (NTHCHARCODE STRING (ADD1 I)))
       do (SETKEYBYTE KEY (SETQ J (IMOD I 8))
                 (LOGXOR (GETKEYBYTE KEY J)
                        (LOGAND (LLSH (PROGN (OR ISCLEAR (SETQ C (\DECRYPT.PWD.CHAR C)))
                                             (COND
                                                [(AND (IGEQ C (CHARCODE A))
                                                      (ILEQ C (CHARCODE Z)))
                                                                   (* Coerce alphabetics to lowercase)
                                                 (IPLUS C (IDIFFERENCE (CHARCODE a)
                                                                       (CHARCODE A]
                                                (T C)))
                                      1)
                                255)))
       finally (RETURN KEY])

)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(DECLARE%: EVAL@COMPILE

(DATATYPE GVKEY ((GVKEY0 8 BYTE)))
)

(/DECLAREDATATYPE 'GVKEY '(BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE)
       ;; ---field descriptor list elided by lister---
       '4)

(DECLARE%: EVAL@COMPILE

(RPAQQ \#BYTES.GVKEY 8)

(CONSTANTS \#BYTES.GVKEY)
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS KEYP MACRO ((X)
                      (type? GVKEY X)))

(PUTPROPS CREATEKEY MACRO (NIL (create GVKEY)))

(PUTPROPS GETKEYBYTE MACRO (= . \GETBASEBYTE))

(PUTPROPS SETKEYBYTE MACRO (= . \PUTBASEBYTE))
)
)

(/DECLAREDATATYPE 'GVKEY '(BYTE BYTE BYTE BYTE BYTE BYTE BYTE BYTE)
       ;; ---field descriptor list elided by lister---
       '4)


          (* * TIMESTAMP)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(DECLARE%: EVAL@COMPILE

[DATATYPE TIMESTAMP ((TIMEHOST BITS 16)
                     (TIMETIMELO WORD)                             (* Mesa numbers backwards)
                     (TIMETIMEHI WORD))
       (ACCESSFNS TIMESTAMP ((TIMETIME (\MAKENUMBER (fetch (TIMESTAMP TIMETIMEHI) of DATUM)
                                                    (fetch (TIMESTAMP TIMETIMELO) of DATUM]
)

(/DECLAREDATATYPE 'TIMESTAMP '((BITS 16)
                               WORD WORD)
       ;; ---field descriptor list elided by lister---
       '4)

(DECLARE%: EVAL@COMPILE

(RPAQQ \#BYTES.TIMESTAMP 6)
```

```
(CONSTANTS \#BYTES.TIMESTAMP)
)
)

(/DECLAREDATATYPE 'TIMESTAMP '((BITS 16)
                               WORD WORD)
        ;; ---field descriptor list elided by lister---
        '4)

(DEFINEQ
```

(\TIMESTAMP.DEFPRINT

```
  [LAMBDA (STAMP STREAM)                                    (* bvm%: "21-May-86 10:44")
    (.SPACECHECK. STREAM 6)
    (\OUTCHAR STREAM (fetch (READTABLEP HASHMACROCHAR) of *READTABLE*))
    (printout STREAM "<GV: " (PORTSTRING (fetch (TIMESTAMP TIMEHOST) of STAMP))
           " at "
           (GDATE (ALTO.TO.LISP.DATE (fetch (TIMESTAMP TIMETIME) of STAMP))
                  (DATEFORMAT NO.SECONDS))
           ">")
    T])
```

(\CHECKSTAMP

```
  [LAMBDA (STAMP)                                           (* bvm%: "19-Jul-85 16:54")
    (COND
       (STAMP (\DTEST STAMP 'TIMESTAMP))
       (T (create TIMESTAMP])

)

(DECLARE%: DONTEVAL@LOAD DOCOPY

(DEFPRINT 'TIMESTAMP '\TIMESTAMP.DEFPRINT)
)


          (* * I/O primitives)

(DEFINEQ
```

(\SENDITEM

```
  [LAMBDA (OUTSTREAM ITEM)                                  (* bvm%: "20-Jul-85 17:30")

          (* send out ITEM as determined by its type as per the specs in section 4.0 of the Grapevine Interface document)

    (COND
       ((FIXP ITEM)
        (\SENDWORD OUTSTREAM ITEM))
       [(OR (LITATOM ITEM)
            (STRINGP ITEM))
        (COND
           (ITEM (\SENDSTRING OUTSTREAM ITEM))
           (T                                               (* not a string at all but an empty string list)
              (\SENDWORD OUTSTREAM 0]
       ((KEYP ITEM)
        (\BOUTS OUTSTREAM ITEM 0 \#BYTES.GVKEY))
       ((type? TIMESTAMP ITEM)
        (\BOUTS OUTSTREAM ITEM 0 \#BYTES.TIMESTAMP))
       [(LISTP ITEM)                                        (* may be a name pair, a string list, or a byte kludge)
        (COND
           [(LITATOM (CDR ITEM))                            (* an RName -
                                                            cons pair of two atoms)
            (LET [(length (IPLUS 1 (NCHARS (CAR ITEM))
                                   (NCHARS (CDR ITEM]
                 (\SENDWORD OUTSTREAM length)
                 (\SENDWORD OUTSTREAM 0)
                 (PRIN3 (CAR ITEM)
                        OUTSTREAM)
                 (BOUT OUTSTREAM (CHARCODE %.))
                 (PRIN3 (CDR ITEM)
                        OUTSTREAM)
                 (COND
                    ((ODDP length)                          (* padding needed)
                     (BOUT OUTSTREAM 0]
           [(EQ (CAR ITEM)
                \3BYTEKLUDGEKEY)

          (* somewhat miss-named now, this gives a way of sending small numbers as bytes instead of words)

            (for b in (CDR ITEM) do (BOUT OUTSTREAM (LOGAND b 255]
           (T                                               (* string list)
              [\SENDWORD OUTSTREAM (for e in ITEM sum (IPLUS 2 (FOLDHI (NCHARS e)
                                                                       BYTESPERWORD]
              (for e in ITEM do (\SENDSTRING OUTSTREAM e]
```

```
       (T (SHOULDNT])
```

(\SENDSTRING

```
  [LAMBDA (STREAM STRING)                                  (* bvm%: "19-Jul-85 16:55")
    (PROG ((L (NCHARS STRING)))
          (COND
             ((IGREATERP L \MAXGVSTRING)
              (ERROR "string too long" STRING)
              (RETURN)))
          (\SENDWORD STREAM L)
          (\SENDWORD STREAM \MAXGVSTRING)                  (* This word is ignored)
          (PRIN3 STRING STREAM)
          (COND
             ((ODDP L)                                     (* pad)
              (BOUT STREAM 0])

)

(DEFINEQ

(\RECEIVEBOOL
  [LAMBDA (STREAM)                                         (* bvm%: "11-MAY-83 14:51")
    (SELECTQ (BIN STREAM)
         (1 T)
         (0 NIL)
         (SHOULDNT])

(\RECEIVECLIST
  [LAMBDA (STREAM)                                         (* bvm%: "11-MAY-83 14:57")
                                                           (* receive a list of components)
    (\RECEIVESTAMP STREAM T)
    (to (\RECEIVEWORD STREAM) collect (\RECEIVECOMPONENT STREAM])

(\RECEIVECOMPONENT
  [LAMBDA (STREAM)                                         (* bvm%: "11-MAY-83 14:57")
                                                           (* receive a component -
                                                           just a list of words)
    (to (\RECEIVEWORD STREAM) collect (\RECEIVEWORD STREAM])

(\RECEIVERLIST
  [LAMBDA (INSTREAM)                                       (* bvm%: "11-MAY-83 15:58")
                                                           (* receive a list of RNames -
                                                           prefix the result with the time STAMP)
    (bind STRLEN (STAMP _ (\RECEIVESTAMP INSTREAM))
          (NWORDS _ (\RECEIVEWORD INSTREAM)) while (IGREATERP NWORDS 0)
       collect (PROG1 (\RECEIVESTRING INSTREAM (SETQ STRLEN (\RECEIVEWORD INSTREAM)))
                                                           (* mind the possible odd length, and add 2 NWORDS for
                                                           STRLEN and max)
                      (SETQ NWORDS (IDIFFERENCE NWORDS (IPLUS (FOLDHI STRLEN BYTESPERWORD)
                                                              2))))
       finally (RETURN (CONS STAMP $$VAL])

(\RECEIVERNAME
  [LAMBDA (INSTREAM)                                       (* bvm%: "11-MAY-83 15:59")
    (\RECEIVESTRING INSTREAM (\RECEIVEWORD INSTREAM])

(\RECEIVESTAMP
  [LAMBDA (STREAM OLDSTAMP)                                (* bvm%: "20-Jul-85 17:16")
    (COND
       ((EQ OLDSTAMP T)
        (RPTQ \#BYTES.TIMESTAMP (BIN STREAM))
        T)
       (T [COND
             ((NOT (type? TIMESTAMP OLDSTAMP))
              (SETQ OLDSTAMP (create TIMESTAMP]
          (\BINS STREAM OLDSTAMP 0 \#BYTES.TIMESTAMP)))
    OLDSTAMP])

(\RECEIVESTRING
  [LAMBDA (STREAM LENGTH)                                  (* bvm%: "21-May-86 10:45")
    (\SKIPWORD STREAM)                                     (* ignore maxLength)
    (LET ((STRING (ALLOCSTRING LENGTH)))
         (\BINS STREAM (fetch (STRINGP BASE) of STRING)
                (fetch (STRINGP OFFST) of STRING)
                LENGTH)
         (COND
            ((ODDP LENGTH)
             (BIN STREAM)))
         STRING])
```

```
)

(RPAQQ \3BYTEKLUDGEKEY $$3byte$$)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(DECLARE%: EVAL@COMPILE

(PUTPROPS \RECEIVEWORD MACRO (= . \WIN))

(PUTPROPS \SKIPWORD MACRO (OPENLAMBDA (STREAM)
                            (PROGN (BIN STREAM)
                                   (BIN STREAM))))

(PUTPROPS \SENDWORD MACRO (= . \WOUT))
)

(DECLARE%: EVAL@COMPILE

(RPAQQ \MAXGVSTRING 64)

(CONSTANTS (\MAXGVSTRING 64))
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \3BYTEKLUDGEKEY)
)
)

(DECLARE%: DONTEVAL@LOAD EVAL@COMPILE DONTCOPY

(SELECTQ (COMPILEMODE)
    (D (FILESLOAD (LOADCOMP)
            PUP BSP))
    (PDP-10 (FILESLOAD (LOADCOMP)
                PUP10 BSPAUX))
    NIL)
)

(PUTPROPS GRAPEVINE COPYRIGHT ("Venue & Xerox Corporation" 1983 1984 1985 1986 1990))
```

FUNCTION INDEX

CONSTANT INDEX

VARIABLE INDEX

MACRO INDEX

RECORD INDEX