File created: 16-May-90 12:10:46 **{DSK}<usr>local>lde>lispcore>sources>AUTHENTICATION.;2**

changes to: (VARS AUTHENTICATIONCOMS)

previous date: 6-Jun-89 11:14:40 {DSK}<usr>local>lde>lispcore>sources>AUTHENTICATION.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
;;
;; Copyright (c) 1987, 1989, 1990 by Venue & Xerox Corporation.  All rights reserved.

(RPAQQ AUTHENTICATIONCOMS
       ((COMS                                                    ; Authentication Protocol
            (COURIERPROGRAMS AUTHENTICATION CHACCESSCONTROL))
        (COMS                                                    ; Strong authentication and changing passwords
            (FNS AS.CHANGE.OWN.PASSWORDS AS.REPLACE.PASSWORDS AS.CREATE.PASSWORDS AS.DELETE.PASSWORDS
                 \AUTHENTICATION.FIND.SERVER AS.MAKE.CONVERSATION AS.NEXT.VERIFIER)
            (ADDVARS (\SYSTEMCACHEVARS \AUTHENTICATION.SERVER.CACHE))
            (VARS AS.WELL.KNOWN.NAME)
            (INITVARS (AUTHENTICATION.NET.HINT)
                   (\AUTHENTICATION.SERVER.CACHE))
            (DECLARE%: DONTCOPY (CONSTANTS (\AUTHENTICATION.SIMPLE.CREDENTIALS 0)
                                           (\AUTHENTICATION.SOCKET 21))
                   (GLOBALVARS AUTHENTICATION.NET.HINT \AUTHENTICATION.SERVER.CACHE AS.WELL.KNOWN.NAME)))
        (COMS                                                    ; Weak authentication
            (FNS NSLOGIN NS.AUTHENTICATE NS.MAKE.SIMPLE.CREDENTIALS HASH.PASSWORD))
        (COMS                                                    ; Clearinghouse access control
            (DECLARE%: EVAL@COMPILE DONTCOPY (FILES (LOADCOMP)
                                                    CLEARINGHOUSE))
            (FNS CH.RETRIEVE.DOMAIN.ACL CH.ADD.MEMBER.TO.DOMAIN.ACL CH.DELETE.MEMBER.FROM.DOMAIN.ACL
                 CH.IS.IN.DOMAIN.ACL CH.RETRIEVE.PROPERTY.ACL CH.ADD.MEMBER.TO.PROPERTY.ACL
                 CH.DELETE.MEMBER.FROM.PROPERTY.ACL CH.NUMBER.TO.PROPERTY))
        (COMS ;; These belong on CLEARINGHOUSE but are here temporarily for benefit of Lyric users wanting a functional NSMAINTAIN.  Put
              ;; these back when a "Lyric" version of this file has been stashed.
            (FNS CH.LIST.PROPERTIES CH.LIST.ORGANIZATIONS CH.LIST.OBJECTS)
            (FNS CH.ADD.GROUP.PROPERTY CH.ADD.MEMBER CH.DELETE.MEMBER))))

;; Authentication Protocol

(COURIERPROGRAM AUTHENTICATION (14 2)
    TYPES
      [(KEY (ARRAY 4 UNSPECIFIED))
       (BLOCK (ARRAY 4 UNSPECIFIED))
       (CREDENTIALS.TYPE (ENUMERATION (SIMPLE 0)
                                      (STRONG 1)))
       [CREDENTIALS (RECORD (TYPE CREDENTIALS.TYPE)
                            (VALUE (SEQUENCE UNSPECIFIED]
       (CREDENTIALS.PACKAGE (RECORD (CREDENTIALS CREDENTIALS)
                                    (NONCE LONGCARDINAL)
                                    (RECIPIENT NSNAME)
                                    (CONVERSATION.KEY KEY)))
       (STRONG.CREDENTIALS (RECORD (CONVERSATION.KEY KEY)
                                   (EXPIRATION.TIME TIME)
                                   (INITIATOR NSNAME)))
       (SIMPLE.CREDENTIALS NSNAME)
       (VERIFIER (SEQUENCE UNSPECIFIED))
       (STRONG.VERIFIER (RECORD (TIMESTAMP TIME)
                                (TICKS LONGCARDINAL)))
       (SIMPLE.VERIFIER HASHED.PASSWORD)
       (HASHED.PASSWORD CARDINAL)
       (PROBLEM (ENUMERATION (CredentialsInvalid 0)
                             (VerifierInvalid 1)
                             (VerifierExpired 2)
                             (VerifierReused 3)
                             (CredentialsExpired 4)
                             (InappropriateCredentials 5)))
       (CALL.PROBLEM (ENUMERATION (TooBusy 0)
                                  (AccessRightsInsufficient 1)
                                  (KeysUnavailable 2)
                                  (StrongKeyDoesNotExist 3)
                                  (SimpleKeyDoesNotExist 4)
                                  (StrongKeyAlreadyRegistered 5)
                                  (SimpleKeyAlreadyRegistered 6)
                                  (DomainForNewKeyUnavailable 7)
                                  (DomainForNewKeyUnknown 8)
                                  (BadKey 9)
                                  (BadName 10)
                                  (DatabaseFull 11)
                                  (Other 12)))
       (WHICH (ENUMERATION (notApplicable 0)
                           (Initiator 1)
```

```
                    (Recipient 2)
                    (Client 3]
    PROCEDURES
      ((BROADCAST.FOR.SERVERS 0 NIL RETURNS ((CLEARINGHOUSE . NETWORK.ADDRESS.LIST)))
       (GET.STRONG.CREDENTIALS 1 (NSNAME NSNAME LONGCARDINAL)
              RETURNS
              ((SEQUENCE UNSPECIFIED))
              REPORTS
              (CALL.ERROR))
       (CHECK.SIMPLE.CREDENTIALS 2 (CREDENTIALS VERIFIER)
              RETURNS
              (BOOLEAN)
              REPORTS
              (AUTHENTICATION.ERROR CALL.ERROR))
       (CREATE.STRONG.KEY 3 (CREDENTIALS VERIFIER NSNAME KEY)
              RETURNS NIL REPORTS (AUTHENTICATION.ERROR CALL.ERROR))
       (CHANGE.STRONG.KEY 4 (CREDENTIALS VERIFIER KEY)
              RETURNS NIL REPORTS (AUTHENTICATION.ERROR CALL.ERROR))
       (DELETE.STRONG.KEY 5 (CREDENTIALS VERIFIER NSNAME)
              RETURNS NIL REPORTS (AUTHENTICATION.ERROR CALL.ERROR))
       (CREATE.SIMPLE.KEY 6 (CREDENTIALS VERIFIER NSNAME HASHED.PASSWORD)
              RETURNS NIL REPORTS (AUTHENTICATION.ERROR CALL.ERROR))
       (CHANGE.SIMPLE.KEY 7 (CREDENTIALS VERIFIER HASHED.PASSWORD)
              RETURNS NIL REPORTS (AUTHENTICATION.ERROR CALL.ERROR))
       (DELETE.SIMPLE.KEY 8 (CREDENTIALS VERIFIER NSNAME)
              RETURNS NIL REPORTS (AUTHENTICATION.ERROR CALL.ERROR)))
    ERRORS
      ((CALL.ERROR 1 (CALL.PROBLEM WHICH))
       (AUTHENTICATION.ERROR 2 (PROBLEM))))

(COURIERPROGRAM CHACCESSCONTROL (127 1)
    INHERITS
      (CLEARINGHOUSE)
    TYPES
      ((DOMAIN.NAME NSNAME2)
       (ORGANIZATION.NAME STRING)
       (WHICH.LIST (ENUMERATION (Readers 0)
                          (valueDONTUSE 1)
                          (Administrators 2)
                          (selfControllers 3)))
       (ELEMENT.NAME NSNAME)
       (DISTING.NAME NSNAME)
       (IS.MEMBER BOOLEAN)
       (ACCESS.LIST (SEQUENCE ELEMENT.NAME))
       (CREDENTIALS (AUTHENTICATION . CREDENTIALS))
       (VERIFIER (AUTHENTICATION . VERIFIER)))
    PROCEDURES
      ((RETRIEVE.PROPERTY.ACL 30 (ELEMENT.NAME PROPERTY WHICH.LIST BULK.DATA.SINK CREDENTIALS VERIFIER)
              RETURNS
              (DISTING.NAME)
              REPORTS
              (CALL.ERROR))
       (ADD.MEMBER.TO.PROPERTY.ACL 31 (ELEMENT.NAME PROPERTY WHICH.LIST ELEMENT.NAME CREDENTIALS VERIFIER)
              RETURNS
              (DISTING.NAME)
              REPORTS
              (CALL.ERROR))
       (DELETE.MEMBER.FROM.PROPERTY.ACL 32 (ELEMENT.NAME PROPERTY WHICH.LIST ELEMENT.NAME CREDENTIALS VERIFIER)
              RETURNS
              (DISTING.NAME)
              REPORTS
              (CALL.ERROR))
       (IS.IN.PROPERTY.ACL 33 (ELEMENT.NAME PROPERTY WHICH.LIST PROPERTY ELEMENT.NAME CREDENTIALS VERIFIER)
              RETURNS
              (IS.MEMBER DISTING.NAME)
              REPORTS
              (CALL.ERROR))
       (RETRIEVE.DOMAIN.ACL 34 (DOMAIN.NAME WHICH.LIST BULK.DATA.SINK CREDENTIALS VERIFIER)
              RETURNS
              (DISTING.NAME)
              REPORTS
              (CALL.ERROR))
       (ADD.MEMBER.TO.DOMAIN.ACL 35 (DOMAIN.NAME WHICH.LIST ELEMENT.NAME CREDENTIALS VERIFIER)
              RETURNS NIL REPORTS (CALL.ERROR))
       (DELETE.MEMBER.FROM.DOMAIN.ACL 36 (DOMAIN.NAME WHICH.LIST ELEMENT.NAME CREDENTIALS VERIFIER)
              RETURNS
              (DISTING.NAME)
              REPORTS
              (CALL.ERROR))
       (IS.IN.DOMAIN.ACL 37 (DOMAIN.NAME WHICH.LIST PROPERTY ELEMENT.NAME CREDENTIALS VERIFIER)
              RETURNS
              (IS.MEMBER)
              REPORTS
              (CALL.ERROR))
       (RETRIEVE.ORGANIZATION.ACL 38 (ORGANIZATION.NAME WHICH.LIST BULK.DATA.SINK CREDENTIALS VERIFIER)
              RETURNS
              (DISTING.NAME)
```

```
            REPORTS
            (CALL.ERROR))
      (ADD.MEMBER.TO.ORGANIZATION.ACL 39 (ORGANIZATION.NAME WHICH.LIST ELEMENT.NAME CREDENTIALS VERIFIER)
            RETURNS
            (DISTING.NAME)
            REPORTS
            (CALL.ERROR))
      (DELETE.MEMBER.FROM.ORGANIZATION.ACL 40 (ORGANIZATION.NAME WHICH.LIST ELEMENT.NAME CREDENTIALS VERIFIER)
            RETURNS
            (DISTING.NAME)
            REPORTS
            (CALL.ERROR))
      (IS.IN.ORGANIZATION.ACL 41 (ORGANIZATION.NAME WHICH.LIST PROPERTY ELEMENT.NAME CREDENTIALS VERIFIER)
            RETURNS
            (IS.MEMBER DISTING.NAME)
            REPORTS
            (CALL.ERROR))))
```

;; Strong authentication and changing passwords

```
(DEFINEQ
```

(AS.CHANGE.OWN.PASSWORDS

```
  [LAMBDA (NEWPASSWORD USERINFO)                          ; Edited 24-Jul-87 16:37 by bvm:
```

;;; Changes user's own password to be NEWPASSWORD, which must be in internal "encrypted" form. USERINFO is the (name . pass) of the user ;;; making/being changed (defaults to global NS user). Returns NIL on success, else an error expression.

```
    (DECLARE (GLOBALVARS AS.WELL.KNOWN.NAME))
    (RESETLST
        (LET ((CONVGOOK (AS.MAKE.CONVERSATION AS.WELL.KNOWN.NAME USERINFO T)))
             (DESTRUCTURING-BIND (STREAM ADDR CREDS . CONVKEY)
                    CONVGOOK
                    (IF (EQ STREAM 'ERROR)
                        THEN                              ; Pass along errors
                             CONVGOOK
                      ELSEIF (COURIER.CALL STREAM 'AUTHENTICATION 'CHANGE.STRONG.KEY CREDS (AS.NEXT.VERIFIER
                                                                                            CONVKEY ADDR)
                                   [DES.BREAKOUT.BLOCKS (CONS (DES.ECB.ENCRYPT CONVKEY (DES.PASSWORD.TO.KEY
                                                                                        NEWPASSWORD]
                                   'RETURNERRORS)
                      ELSEIF (COURIER.CALL STREAM 'AUTHENTICATION 'CHANGE.SIMPLE.KEY CREDS (AS.NEXT.VERIFIER
                                                                                            CONVKEY ADDR)
                                   (HASH.PASSWORD NEWPASSWORD)
                                   'RETURNERRORS)
                      ELSE                                ; Success if neither call returned an error
                           T))))])
```

(AS.REPLACE.PASSWORDS

```
  [LAMBDA (NAME NEWPASSWORD)                              ; Edited 22-Jul-87 17:37 by bvm:
```

;; Replace the strong and simple keys for user NAME with NEWPASSWORD. This requires deleting the old keys and creating new ones.

```
    (DECLARE (GLOBALVARS AS.WELL.KNOWN.NAME))
    (SETQ NAME (PARSE.NSNAME NAME))
    (RESETLST
        (LET ((CONVGOOK (AS.MAKE.CONVERSATION AS.WELL.KNOWN.NAME NIL T))
              ERROR)
             (DESTRUCTURING-BIND (STREAM ADDR CREDS . CONVKEY)
                    CONVGOOK
                    (if (EQ STREAM 'ERROR)
                        then                              ; Pass along errors
                             CONVGOOK
                      elseif (AND (SETQ ERROR (COURIER.CALL STREAM 'AUTHENTICATION 'DELETE.STRONG.KEY CREDS
                                                  (AS.NEXT.VERIFIER CONVKEY ADDR)
                                                  NAME
                                                  'RETURNERRORS))
                                  (NEQ (CADDR ERROR)
                                       'StrongKeyDoesNotExist))
                        THEN                              ; Don't complain if strong key doesn't exist to delete
                             ERROR
                      elseif (AND (SETQ ERROR (COURIER.CALL STREAM 'AUTHENTICATION 'DELETE.SIMPLE.KEY CREDS
                                                  (AS.NEXT.VERIFIER CONVKEY ADDR)
                                                  NAME
                                                  'RETURNERRORS))
                                  (NEQ (CADDR ERROR)
                                       'SimpleKeyDoesNotExist))
                        THEN ERROR
                      elseif (COURIER.CALL STREAM 'AUTHENTICATION 'CREATE.STRONG.KEY CREDS (AS.NEXT.VERIFIER
                                                                                            CONVKEY ADDR)
                                   NAME
                                   [DES.BREAKOUT.BLOCKS (CONS (DES.ECB.ENCRYPT CONVKEY (DES.PASSWORD.TO.KEY
                                                                                        NEWPASSWORD]
                                   'RETURNERRORS)
                      ELSEIF (COURIER.CALL STREAM 'AUTHENTICATION 'CREATE.SIMPLE.KEY CREDS (AS.NEXT.VERIFIER
                                                                                            CONVKEY ADDR)
                                    NAME
```

```
                                (HASH.PASSWORD NEWPASSWORD)
                                'RETURNERRORS)
                    else                                        ; Success if neither call returned an error
                         T)))))])
```

(AS.CREATE.PASSWORDS

```
[LAMBDA (NAME PASSWORD)                                         ; Edited 30-Jul-87 17:25 by bvm:
```

;;; Create Strong and Simple keys for user NAME. PASSWORD is in the "encrypted" form used by \INTERNAL/GETPASSWORD.

```
    (DECLARE (GLOBALVARS AS.WELL.KNOWN.NAME))
    (SETQ NAME (PARSE.NSNAME NAME))
    (RESETLST
        (LET ((CONVGOOK (AS.MAKE.CONVERSATION AS.WELL.KNOWN.NAME NIL T)))
             (DESTRUCTURING-BIND (STREAM ADDR CREDS . CONVKEY)
                    CONVGOOK
                    (IF (EQ STREAM 'ERROR)
                        THEN                                    ; Pass along errors
                             CONVGOOK
                      ELSEIF (COURIER.CALL STREAM 'AUTHENTICATION 'CREATE.STRONG.KEY CREDS (AS.NEXT.VERIFIER
                                                                                            CONVKEY ADDR)
                                  NAME
                                  [DES.BREAKOUT.BLOCKS (CONS (DES.ECB.ENCRYPT CONVKEY (DES.PASSWORD.TO.KEY
                                                                                       PASSWORD]
                                  'RETURNERRORS)
                      ELSEIF (COURIER.CALL STREAM 'AUTHENTICATION 'CREATE.SIMPLE.KEY CREDS (AS.NEXT.VERIFIER
                                                                                            CONVKEY ADDR)
                                  NAME
                                  (HASH.PASSWORD PASSWORD)
                                  'RETURNERRORS)
                      ELSE                                      ; Success if neither call returned an error
                           T)))))])
```

(AS.DELETE.PASSWORDS

```
[LAMBDA (NAME)                                                  ; Edited 22-Jul-87 13:07 by bvm:
```

;; Delete the strong and simple keys for user NAME

```
    (DECLARE (GLOBALVARS AS.WELL.KNOWN.NAME))
    (SETQ NAME (PARSE.NSNAME NAME))
    (RESETLST
        (LET ((CONVGOOK (AS.MAKE.CONVERSATION AS.WELL.KNOWN.NAME NIL T)))
             (DESTRUCTURING-BIND (STREAM ADDR CREDS . CONVKEY)
                    CONVGOOK
                    (if (EQ STREAM 'ERROR)
                        then                                    ; Pass along errors
                             CONVGOOK
                      elseif (COURIER.CALL STREAM 'AUTHENTICATION 'DELETE.STRONG.KEY CREDS (AS.NEXT.VERIFIER
                                                                                            CONVKEY ADDR)
                                  NAME
                                  'RETURNERRORS)
                      elseif (COURIER.CALL STREAM 'AUTHENTICATION 'DELETE.SIMPLE.KEY CREDS (AS.NEXT.VERIFIER
                                                                                            CONVKEY ADDR)
                                  NAME
                                  'RETURNERRORS)
                      else                                      ; Success if neither call returned an error
                           T)))))])
```

(\AUTHENTICATION.FIND.SERVER

```
[LAMBDA NIL                                                     (* bvm%: " 1-Jul-84 15:26")
```

;; Expanding ring broadcast, as defined in Clearinghouse Protocol spec.

```
    (PROG (INFO)
          (RETURN (COND
                     [(AND \AUTHENTICATION.SERVER.CACHE (find ADDR in \AUTHENTICATION.SERVER.CACHE
                                                        suchthat (SELECTQ [CAR (LISTP (COURIER.EXPEDITED.CALL
                                                                                       ADDR
                                                                                       \AUTHENTICATION.SOCKET
                                                                                       'AUTHENTICATION
                                                                                       'BROADCAST.FOR.SERVERS
                                                                                       'RETURNERRORS]
                                                                          ((NIL ERROR REJECT)
                                                                              NIL)
                                                                          T]
                     ((SETQ INFO (COURIER.BROADCAST.CALL \AUTHENTICATION.SOCKET 'AUTHENTICATION
                                                        'BROADCAST.FOR.SERVERS NIL NIL AUTHENTICATION.NET.HINT
                                                        "Authentication servers"))
                      (SETQ \AUTHENTICATION.SERVER.CACHE (APPEND INFO \AUTHENTICATION.SERVER.CACHE))
                      (CAR INFO)])
```

(AS.MAKE.CONVERSATION

```
[LAMBDA (RECIPIENT USERINFO KEEPSTREAM)                         ; Edited 23-Jul-87 10:44 by bvm:
```

;; Set up a conversation with RECIPIENT by obtaining strong credentials. If USERINFO is supplied, it is a (user . password) pair, defaults to the
;; global NS login. Value returned is (credentials . conversationkey). If KEEPSTREAM is true, then the caller plans to converse with an

```
    ;; authentication service, in which case the same courier stream can be used here and there, and the value returned is (stream address credentials
    ;; . conversationkey).  Caller needs a resetlst for this option.
    [OR USERINFO (SETQ USERINFO (\INTERNAL/GETPASSWORD '|NS::|]
    (LET ((NONCE (RAND))
          CRED.PACK ADDR STREAM)
         (if [AND (SETQ ADDR (\AUTHENTICATION.FIND.SERVER))
                  (OR (NULL KEEPSTREAM)
                      (SETQ STREAM (COURIER.OPEN ADDR NIL T 'AUTHENTICATION]
             then (if KEEPSTREAM
                      then (RESETSAVE NIL (LIST (FUNCTION \SPP.RESETCLOSE)
                                                STREAM)))
                  (COND
                     ((EQ [CAR (SETQ CRED.PACK (COURIER.CALL (OR STREAM ADDR)
                                                             'AUTHENTICATION
                                                             'GET.STRONG.CREDENTIALS
                                                             (PARSE.NSNAME (CAR USERINFO))
                                                             (PARSE.NSNAME RECIPIENT)
                                                             NONCE
                                                             'RETURNERRORS]
                          'ERROR)                                    ; Return error
                      CRED.PACK)
                     ((OR [NULL (NLSETQ                              ; If our key is wrong, the decoding could break as we try to read
                                                                     ; a garbage sequence
                                  (SETQ CRED.PACK (COURIER.READ.REP (DES.BREAKOUT.BLOCKS
                                                                       (DES.CBCC.DECRYPT (DES.PASSWORD.TO.KEY
                                                                                          (CDR USERINFO))
                                                                                         (DES.MAKE.BLOCKS CRED.PACK)))
                                                                     'AUTHENTICATION
                                                                     'CREDENTIALS.PACKAGE]
                          (NOT (IEQP (COURIER.FETCH (AUTHENTICATION . CREDENTIALS.PACKAGE)
                                                    NONCE of CRED.PACK)
                                     NONCE)))                        ; decoding failed--either our key is wrong or the authentication
                                                                     ; server is bogus.  We assume the latter is unlikely, so report bad
                                                                     ; key
                      '(ERROR AUTHENTICATION.ERROR CredentialsInvalid))
                     (T [SETQ CRED.PACK (CONS (COURIER.FETCH (AUTHENTICATION . CREDENTIALS.PACKAGE)
                                                             CREDENTIALS of CRED.PACK)
                                              (DES.MAKE.KEY (COURIER.FETCH (AUTHENTICATION
                                                                            . CREDENTIALS.PACKAGE)
                                                                           CONVERSATION.KEY of CRED.PACK]
                        (if KEEPSTREAM
                            then (CONS STREAM (CONS ADDR CRED.PACK))
                            else CRED.PACK)))
             else '(ERROR CALL.ERROR Can'tGetAuthenticationService])
```

(AS.NEXT.VERIFIER

```
  [LAMBDA (CONVKEY ADDR)                                             (* jwo%: " 9-Aug-85 01:50")
```

;;; The long garbage in the IF is and attempt to XOR the recipients 'processor id' with the courier data representation, before encrypting.

```
    (DES.BREAKOUT.BLOCKS
     (LET [(BL (DES.MAKE.BLOCKS (LET [(L (COURIER.WRITE.REP (COURIER.CREATE (AUTHENTICATION . STRONG.VERIFIER)
                                                                            TIMESTAMP _ (IDATE)
                                                                            TICKS _ (RAND))
                                                            'AUTHENTICATION
                                                            'STRONG.VERIFIER]
                                                      [if (CAR L)
                                                          then (RPLACA L (LOGXOR (CAR L)
                                                                                 (fetch (NSADDRESS NSHNM0) of ADDR)))
                                                               (if (CADR L)
                                                                   then (RPLACA (CDR L)
                                                                                (LOGXOR (CADR L)
                                                                                        (fetch (NSADDRESS NSHNM1) of ADDR)))
                                                                        (if (CADDR L)
                                                                            then (RPLACA (CDDR L)
                                                                                         (LOGXOR (CADDR L)
                                                                                                 (fetch (NSADDRESS NSHNM2)
                                                                                                    of ADDR]
                                                      L]
           (for E in BL collect (DES.ECB.ENCRYPT CONVKEY E])

)

(ADDTOVAR \SYSTEMCACHEVARS \AUTHENTICATION.SERVER.CACHE)

(RPAQQ AS.WELL.KNOWN.NAME "Authentication Service:CHServers:CHServers")

(RPAQ? AUTHENTICATION.NET.HINT )

(RPAQ? \AUTHENTICATION.SERVER.CACHE )

(DECLARE%: DONTCOPY

(DECLARE%: EVAL@COMPILE

(RPAQQ \AUTHENTICATION.SIMPLE.CREDENTIALS 0)
```

```
(RPAQQ \AUTHENTICATION.SOCKET 21)

(CONSTANTS (\AUTHENTICATION.SIMPLE.CREDENTIALS 0)
       (\AUTHENTICATION.SOCKET 21))
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS AUTHENTICATION.NET.HINT \AUTHENTICATION.SERVER.CACHE AS.WELL.KNOWN.NAME)
)
)
```

;; Weak authentication

```
(DEFINEQ
```

(NSLOGIN

```
  [LAMBDA (HOST MSG)                                                    (* bvm%: "23-Aug-84 15:10")
    (\INTERNAL/GETPASSWORD HOST T NIL MSG NIL 'NS])
```

(NS.AUTHENTICATE

```
  [LAMBDA (SIMPLE.CREDENTIALS)                                          (* bvm%: "15-Aug-84 16:00")
```

;;; Checks SIMPLE.CREDENTIALS -- For convenience, if SIMPLE.CREDENTIALS is not a list, creates credentials from the login for NS::

```
    [OR (LISTP SIMPLE.CREDENTIALS)
        (SETQ SIMPLE.CREDENTIALS (NS.MAKE.SIMPLE.CREDENTIALS (\INTERNAL/GETPASSWORD '|NS::| SIMPLE.CREDENTIALS]
    (PROG ((ADDR (\AUTHENTICATION.FIND.SERVER))
           RESULT)
          (RETURN (COND
                     ((NULL ADDR)
                      'AllDown)
                     (T (SETQ RESULT (COURIER.CALL ADDR 'AUTHENTICATION 'CHECK.SIMPLE.CREDENTIALS (CAR
                                                                                                    SIMPLE.CREDENTIALS
                                                                                                           )
                                                                    (CDR SIMPLE.CREDENTIALS)
                                                                    'RETURNERRORS))
                        (COND
                           ((LISTP RESULT)
                            (CADDR RESULT))
                           (RESULT)
                           (T 'CredentialsInvalid])
```

(NS.MAKE.SIMPLE.CREDENTIALS

```
  [LAMBDA (NAME/PASS)                                                   (* bvm%: "15-Aug-84 15:30")
    (CONS (COURIER.CREATE (AUTHENTICATION . CREDENTIALS)
                  TYPE _ 'SIMPLE VALUE _ (COURIER.WRITE.REP (PARSE.NSNAME (CAR NAME/PASS))
                                                            'AUTHENTICATION
                                                            'SIMPLE.CREDENTIALS))
          (COURIER.WRITE.REP (HASH.PASSWORD (CDR NAME/PASS))
                  'AUTHENTICATION
                  'SIMPLE.VERIFIER])
```

(HASH.PASSWORD

```
  [LAMBDA (PASSWORD)                                                    (* bvm%: " 3-NOV-83 22:35")
```

;; Compute remainder mod 65357 of PASSWORD considered as an arbitrary length integer whose 16 bit words, from most to least significant, are
;; the characters in PASSWORD. Uses Horner's rule and properties of modular arithmetic to do it efficiently.

```
    (bind (HASH _ 0) for CHAR instring (MKSTRING PASSWORD)
       do (SETQ HASH (IMOD (IPLUS (ITIMES HASH (CONSTANT (IMOD (EXPT 2 16)
                                                               65357)))
                                  (L-CASECODE (\DECRYPT.PWD.CHAR CHAR)))
                           65357))
       finally (RETURN HASH])

)
```

;; Clearinghouse access control

```
(DECLARE%: EVAL@COMPILE DONTCOPY

(FILESLOAD (LOADCOMP)
       CLEARINGHOUSE)
)

(DEFINEQ
```

(CH.RETRIEVE.DOMAIN.ACL

```
  [LAMBDA (DOMAIN WHICH.LIST)                                           (* jwo%: "24-Jun-85 14:54")
    (LET ((AUTH (CH.GETAUTHENTICATOR T)))
         (COURIER.CALL (CH.FINDSERVER (SETQ DOMAIN (PARSE.NSNAME DOMAIN 2)))
                'CHACCESSCONTROL
```

```
               'RETRIEVE.DOMAIN.ACL DOMAIN WHICH.LIST '(CHACCESSCONTROL . ELEMENT.NAME)
               (COURIER.FETCH (CLEARINGHOUSE . AUTHENTICATOR)
                      CREDENTIALS of AUTH)
               (COURIER.FETCH (CLEARINGHOUSE . AUTHENTICATOR)
                      VERIFIER of AUTH)
               'RETURNERRORS])
```

(CH.ADD.MEMBER.TO.DOMAIN.ACL

```
(LAMBDA (DOMAIN WHICH.LIST NEWMEMBER DONTCHECK)                ; Edited 10-Aug-87 14:54 by bvm:
  (OR DONTCHECK (SETQ NEWMEMBER (CH.CANONICAL.NAME NEWMEMBER)))
  (LET ((AUTH (CH.GETAUTHENTICATOR T)))
       (COURIER.EXPEDITED.CALL (CH.FINDSERVER (SETQ DOMAIN (PARSE.NSNAME DOMAIN 2)))
              \CH.BROADCAST.SOCKET
              'CHACCESSCONTROL
              'ADD.MEMBER.TO.DOMAIN.ACL DOMAIN WHICH.LIST NEWMEMBER (COURIER.FETCH (CLEARINGHOUSE
                                                                                    . AUTHENTICATOR)
                                                                          CREDENTIALS of AUTH)
              (COURIER.FETCH (CLEARINGHOUSE . AUTHENTICATOR)
                      VERIFIER of AUTH)
              'RETURNERRORS])
```

(CH.DELETE.MEMBER.FROM.DOMAIN.ACL

```
(LAMBDA (DOMAIN WHICH.LIST OLDMEMBER DONTCHECK)                ; Edited 10-Aug-87 14:55 by bvm:
  (OR DONTCHECK (SETQ OLDMEMBER (CH.CANONICAL.NAME OLDMEMBER)))
  (LET ((AUTH (CH.GETAUTHENTICATOR T)))
       (COURIER.EXPEDITED.CALL (CH.FINDSERVER (SETQ DOMAIN (PARSE.NSNAME DOMAIN 2)))
              \CH.BROADCAST.SOCKET
              'CHACCESSCONTROL
              'DELETE.MEMBER.FROM.DOMAIN.ACL DOMAIN WHICH.LIST OLDMEMBER (COURIER.FETCH (CLEARINGHOUSE
                                                                                       . AUTHENTICATOR)
                                                                             CREDENTIALS of AUTH)
              (COURIER.FETCH (CLEARINGHOUSE . AUTHENTICATOR)
                      VERIFIER of AUTH)
              'RETURNERRORS])
```

(CH.IS.IN.DOMAIN.ACL

```
(LAMBDA (DOMAIN WHICH PROPERTY NAME)                           (* jwo%: " 9-Aug-85 18:55")
  (LET ((AUTH (CH.GETAUTHENTICATOR T)))
       (COURIER.CALL (CH.FINDSERVER (SETQ DOMAIN (PARSE.NSNAME DOMAIN 2))
                            T)
              'CHACCESSCONTROL
              'IS.IN.DOMAIN.ACL DOMAIN WHICH (OR (CH.PROPERTY PROPERTY)
                                                 PROPERTY)
              (PARSE.NSNAME NAME)
              (COURIER.FETCH (CLEARINGHOUSE . AUTHENTICATOR)
                      CREDENTIALS of AUTH)
              (COURIER.FETCH (CLEARINGHOUSE . AUTHENTICATOR)
                      VERIFIER of AUTH)
              'RETURNERRORS])
```

(CH.RETRIEVE.PROPERTY.ACL

```
(LAMBDA (NAME PROPERTY WHICH.LIST)                             (* jwo%: "24-Jun-85 14:37")
  (LET ((AUTH (CH.GETAUTHENTICATOR T)))
       (COURIER.CALL (CH.FINDSERVER (SETQ NAME (PARSE.NSNAME NAME))
                            T)
              'CHACCESSCONTROL
              'RETRIEVE.PROPERTY.ACL NAME (OR (CH.PROPERTY PROPERTY)
                                              PROPERTY)
              WHICH.LIST
              '(CHACCESSCONTROL . ELEMENT.NAME)
              (COURIER.FETCH (CLEARINGHOUSE . AUTHENTICATOR)
                      CREDENTIALS of AUTH)
              (COURIER.FETCH (CLEARINGHOUSE . AUTHENTICATOR)
                      VERIFIER of AUTH)
              'RETURNERRORS])
```

(CH.ADD.MEMBER.TO.PROPERTY.ACL

```
(LAMBDA (OBJECT PROPERTY WHICH.LIST NEWMEMBER DONTCHECK)       ; Edited 10-Aug-87 14:55 by bvm:
  (OR DONTCHECK (SETQ NEWMEMBER (CH.CANONICAL.NAME NEWMEMBER)))
  (SETQ OBJECT (PARSE.NSNAME OBJECT))
  (LET ((AUTH (CH.GETAUTHENTICATOR T)))
       (COURIER.CALL (CH.FINDSERVER OBJECT)
              'CHACCESSCONTROL
              'ADD.MEMBER.TO.PROPERTY.ACL OBJECT (OR (CH.PROPERTY PROPERTY)
                                                     PROPERTY)
              WHICH.LIST
              (PARSE.NSNAME NEWMEMBER)
              (COURIER.FETCH (CLEARINGHOUSE . AUTHENTICATOR)
                      CREDENTIALS of AUTH)
              (COURIER.FETCH (CLEARINGHOUSE . AUTHENTICATOR)
                      VERIFIER of AUTH)
              'RETURNERRORS])
```

(CH.DELETE.MEMBER.FROM.PROPERTY.ACL

```
  [LAMBDA (OBJECT PROPERTY WHICH.LIST OLDMEMBER DONTCHECK)       ; Edited 10-Aug-87 15:44 by bvm:
    (OR DONTCHECK (SETQ OLDMEMBER (CH.CANONICAL.NAME OLDMEMBER)))
    (SETQ OBJECT (PARSE.NSNAME OBJECT))
    (LET ((AUTH (CH.GETAUTHENTICATOR T)))
         (COURIER.CALL (CH.FINDSERVER OBJECT)
                'CHACCESSCONTROL
                'DELETE.MEMBER.FROM.PROPERTY.ACL OBJECT (OR (CH.PROPERTY PROPERTY)
                                                            PROPERTY)
                WHICH.LIST
                (PARSE.NSNAME OLDMEMBER)
                (COURIER.FETCH (CLEARINGHOUSE . AUTHENTICATOR)
                       CREDENTIALS of AUTH)
                (COURIER.FETCH (CLEARINGHOUSE . AUTHENTICATOR)
                       VERIFIER of AUTH)
                'RETURNERRORS])
```

(CH.NUMBER.TO.PROPERTY

```
  [LAMBDA (PNUM)                                                 (* ejs%: "10-Jun-85 16:26")

;;; reverse mapping to that of CH.PROPERTY

    (CAR (for M in CH.PROPERTIES thereis (EQ PNUM (CADR M])

)

;; These belong on CLEARINGHOUSE but are here temporarily for benefit of Lyric users wanting a functional NSMAINTAIN.  Put these back when a
;; "Lyric" version of this file has been stashed.

(DEFINEQ
```

(CH.LIST.PROPERTIES

```
  [LAMBDA (OBJECTNAMEPATTERN)                                    ; Edited 24-Jul-87 15:38 by bvm:
    (COURIER.EXPEDITED.CALL (CH.FINDSERVER (SETQ OBJECTNAMEPATTERN (PARSE.NSNAME OBJECTNAMEPATTERN)))
           \CH.BROADCAST.SOCKET
           'CLEARINGHOUSE
           'LIST.PROPERTIES OBJECTNAMEPATTERN (CH.GETAUTHENTICATOR)
           'RETURNERRORS])
```

(CH.LIST.ORGANIZATIONS

```
  [LAMBDA (ORGANIZATIONPATTERN)                                  ; Edited 24-Jul-87 17:47 by bvm:
    (COURIER.CALL (GETCLEARINGHOUSE)
           'CLEARINGHOUSE
           'LIST.ORGANIZATIONS
           (PARSE.NSNAME ORGANIZATIONPATTERN 1)
           '(CLEARINGHOUSE . ORGANIZATION)
           (CH.GETAUTHENTICATOR)
           'RETURNERRORS])
```

(CH.LIST.OBJECTS

```
  [LAMBDA (OBJECTPATTERN PROPERTY)                               ; Edited 24-Jul-87 17:47 by bvm:
    (COURIER.CALL (CH.FINDSERVER (SETQ OBJECTPATTERN (PARSE.NSNAME OBJECTPATTERN))
                          T)
           'CLEARINGHOUSE
           'LIST.OBJECTS OBJECTPATTERN (CH.PROPERTY (OR PROPERTY 'ALL))
           '(CLEARINGHOUSE . OBJECT)
           (CH.GETAUTHENTICATOR)
           'RETURNERRORS])

)

(DEFINEQ
```

(CH.ADD.GROUP.PROPERTY

```
  [LAMBDA (OBJECTNAME PROPERTY MEMBERS DONTCHECK)                ; Edited 10-Aug-87 14:57 by bvm:
    [OR DONTCHECK (SETQ MEMBERS (for X in MEMBERS collect (CH.CANONICAL.NAME X]
    (COURIER.CALL (CH.FINDSERVER (SETQ OBJECTNAME (PARSE.NSNAME OBJECTNAME)))
           'CLEARINGHOUSE
           'ADD.GROUP.PROPERTY OBJECTNAME (OR (FIXP PROPERTY)
                                              (CH.PROPERTY PROPERTY))
           [FUNCTION (LAMBDA (DATASTREAM)                        ; Function to write the membership onto the bulk data stream
                     (COURIER.WRITE.BULKDATA DATASTREAM MEMBERS NIL 'NSNAME]
           (CH.GETAUTHENTICATOR T)
           'RETURNERRORS])
```

(CH.ADD.MEMBER

```
  [LAMBDA (GROUPNAME PROPERTY NEWMEMBER DONTCHECK)               ; Edited 10-Aug-87 14:51 by bvm:
    (OR DONTCHECK (SETQ NEWMEMBER (CH.CANONICAL.NAME NEWMEMBER)))
    (COURIER.CALL (CH.FINDSERVER (SETQ GROUPNAME (PARSE.NSNAME GROUPNAME)))
           'CLEARINGHOUSE
           'ADD.MEMBER GROUPNAME (OR (FIXP PROPERTY)
```

```
                                    (CH.PROPERTY PROPERTY))
           NEWMEMBER
           (CH.GETAUTHENTICATOR T)
           'RETURNERRORS])
```

(CH.DELETE.MEMBER

```
  [LAMBDA (GROUPNAME PROPERTY OLDMEMBER DONTCHECK)              ; Edited 10-Aug-87 14:50 by bvm:
    (OR DONTCHECK (SETQ OLDMEMBER (CH.CANONICAL.NAME OLDMEMBER)))
    (COURIER.CALL (CH.FINDSERVER (SETQ GROUPNAME (PARSE.NSNAME GROUPNAME)))
           'CLEARINGHOUSE
           'DELETE.MEMBER GROUPNAME (OR (FIXP PROPERTY)
                                        (CH.PROPERTY PROPERTY))
           OLDMEMBER
           (CH.GETAUTHENTICATOR T)
           'RETURNERRORS])

)

(PUTPROPS AUTHENTICATION COPYRIGHT ("Venue & Xerox Corporation" 1987 1989 1990))
```

FUNCTION INDEX

VARIABLE INDEX

CONSTANT INDEX

COURIERPROGRAM INDEX