ifdef __STDC__
PUBLIC int HTInetStatus(char *where)
#else
PUBLIC int HTInetStatus(where)
    char *where;
#endif
{
    CTRACE(tfp, "TCP: Error %d in 'errno' after call to %s() failed.\n\t%*s\n", errno, where,
    #ifdef VM
    "(Error number not translated)") ;    /* What Is the VM equiv? */
    #define ER_NO_TRANS_DONE
    #endif
    #ifdef vms
    "(Error number not translated)") ;
    #define ER_NO_TRANS_DONE
    #endif
    #ifdef NeXT
    strerror(errno));
    #define ER_NO_TRANS_DONE
    #endif

    ifndef ER_NO_TRANS_DONE
    errno < sys_nerr ? sys_errlist[errno] : "Unknown error" );
    #endif

    #ifdef vms
    CTRACE(tfp, " Unix error number (uerrno) = %ld dec\n", uerrno);
    CTRACE(tfp, " VMS error (vmserro) = %lx hex\n", vmserro);
    #endif
    return -errno;

/*
 ** Parse a cardinal value
 **-----------------------
 ** On entry,
 **  *pp         points to first character to be interpreted, terminated by
 **  *pstatus    non 0:9 character.
 **  *maxvalue   points to status already valid
 **  gives the largest allowable value.
 ** On exit,
 **  *pp         points to first unread character
 **  *pstatus    points to status updated iff bad
 */
#ifdef __STDC__
PUBLI C unsigned int HTCardinal(int *pstatus, char **pp,
    unsigned int max_value)
#else
PUBLI C unsigned int HTCardinal(pstatus, pp, max_value)
    int *pstatus;
    char **pp;
    unsigned int max_value;
#endif
{
    int n;
if ( (**pp< '0') || (**pp> '9')) { /* Null string is error */
    *pstatus = -3; /* No number where one expected */
    return 0;
}

n=0;
while ( (**pp>='0') && (**pp<='9')) n = n*10 + (*((pp)++)) - '0';

if (n>max_value) {
    *pstatus = -4; /* Cardinal outside range */
    return 0;
}

return n;

/*
 * Produce a string for an internet address
 */

** On exit,
 * returns a pointer to a static string which must be copied if
 * it is to be kept.
 */
#endif
PUBLIC const char * HTInetString(struct sockaddr_in* sin)
#else
PUBLIC char * HTInetString(sin)
   struct sockaddr_in *sin;
#endif
{
    static char string[16];
    sprintf(string, "%d.%d.%d.%d",
        (int)*((unsigned char *)(&HTHostAddress.sin_addr) +0),
        (int)*((unsigned char *)(&HTHostAddress.sin_addr) +1),
        (int)*((unsigned char *)(&HTHostAddress.sin_addr) +2),
        (int)*((unsigned char *)(&HTHostAddress.sin_addr) +3));
    return string;
}

/*
 * Parse an internet node address and port
 */

** On entry,
 * str points to a string with a node name or number,
 * with optional trailing colon and port number.
 * sin points to the binary internet address field.
 **
 * On exit,
 * *sin is filled in. If no port is specified in str, that
 * field is left unchanged in *sin.
 */
#endif
PUBLIC int HTParseInet(struct sockaddr_in* sin, const char *str)
#else
PUBLIC int HTParseInet(sin, str)
   struct sockaddr_in *sin,
   char *str;
#endif
{ 
    char *port;
    char host[256];
    struct hostent *hostp; /* Pointer to host - See netdb.h */
    strcpy(host, str); /* Take a copy we can mutilate */

    /* Parse port number if present */
    if (port=strchr(host, ':')) {
        *port++ = 0; /* Chop off port */
        if (port[0]>'0' && port[0]<'9') {
            sin->sin_port = htons(strtol(port, (char**)&0, 10));
        } else if (TRACE) /* Crashes!?!  Not recommended */
            struct servent *serv = getservbyname(port, (char*)0);
            if (serv) sin->sin_port = serv->s_port;
            else if (TRACE) printf("TCP: Unknown service \%

        ");
    
    /* Parse host number if present */
    if (*host>'0' && *host<'9') { /* Numeric node address: */
        sin->sin_addr.s_addr = inet_addr(host); /* See arpa/inet.h */
    } else if (TRACE) printf("TCP: Parsed address as port \%

        ");

    */

    if (TRACE) printf("TCP: Parsed address as port \%

        ");

    if (!host) {
        if (TRACE) printf("TCP: host not found \%

        ");
    }

    return 0; /* OK */
}

/* Derive the name of the host on which we are */

/ * IIedef __STDC__
 #ifdef MAXHOSTNAMELEN
 #define MAXHOSTNAMELEN 64
 #endif

 { char name[MAXHOSTNAMELEN+1];
 struct hostent * phost;
 int namelength = sizeof(name);

 if (hostname) return; /* Already done */
 gethostbyname(name, namelength); /* Without domain */
 CTRACE(tfp, "TCP: Local host name is %s\n", name);
 phost=gethostbyname(name); /* See netdb.h */
 if (!phost) {
   if (TRACE) printf("TCP: Can't find my own internet node address for '%s'!!\n", name);
   return; /* Fail! */
 }
 StrAllocCopy(hostname, phost->h_name);
 memcpy(&HTHostName, &phost->h_addr, phost->h_length);
 if (TRACE) printf(" Name server says that is '%s' = %s\n", hostname, HTInetString(&HTHostName));
 }

 #ifdef __STDC__
 PUBLIC const char * HTHostName(void)
 #else
 PUBLIC char * HTHostName()
 #endif
 {
   get_host_details();
   return hostname;
}
Generic Communication Code

Requires tcp.h to be included first.

Produce a string for an internet address

On exit,
returns a pointer to a static string which must be copied if
it is to be kept.

Encode INET status (as in sys/errno.h)

On entry,
gives a description of what caused the error

gives the error number in the unix way.

On return,
returns
a negative status in the unix way.

Publicly accessible variables

extern struct sockaddr_in HTHostAddress; /* The internet address of the host */
   /* Valid after call to HTHostName() */

Parse a cardinal value

On entry,
points to first character to be interpreted, terminated by
non 0:9 character.

points to status already valid

gives the largest allowable value.

On exit,
points to first unread character

points to status updated iff bad

8/29/01
extern unsigned int HTCardinal(int *pstatus,
    char **pp,
    unsigned int max_value);
#else
extern unsigned int HTCardinal();
#endif

/* Parse an internet node address and port */
/* ------------------------------------------
 ** On entry, 
 ** str points to a string with a node name or number, 
 ** with optional trailing colon and port number. 
 ** sin points to the binary internet address field. 
 ** On exit, 
 ** *sin is filled in. If no port is specified in str, that 
 ** field is left unchanged in *sin. */
#ifdef __STDC__
extern int HTParseInet(struct sockaddr_in* sin, const char *str);
#else
extern int HTParseInet();
#endif

/* Get Name of This Machine */
/* --------------------------
 */
#ifdef __STDC__
extern const char * HTHostName(void);
#else
extern char * HTHostName();
#endif
/* Macros for general use */

/* Allocate a new copy of a string: */
#define StrAllocCopy(new,old) \
    if (new) free(new); (new) = (char *)malloc(strlen(old)+1); strcpy((new),(old))
/* extern void *malloc(size_t size); */

/* Debug message control. */
#ifdef DEBUG
#define TRACE (WWW_TraceFlag)
#define PROGRESS(str) printf(str)
extern int WWW_TraceFlag;
#else
#define TRACE 0
#define PROGRESS(str) /* nothing for now */
#endif
#define CTRACE if(TRACE)fprintf
#define tfp stdout

/* Standard C library for malloc() etc */
#ifdef vax
#endif
#ifdef unix
#define ultrix /* Assume vax+unix=ultrix */
#endif
#endif

#ifndef VMS
#ifndef ultrix
#ifndef NeXT
#include <libc.h> /* NeXT */
#endif
#include <stdlib.h> /* ANSI */
#else
#include <malloc.h> /* ultrix */
#endif
#include <memory.h>
#endif
#else
#ifndef stdio.h>
#include <stdio.h>
#else
#include <ctype.h>
#endif

#define PUBLIC /* Accessible outside this module */
#define PRIVATE static /* Accessible only within this module */

#ifdef _STDC_
#define CONST const /* "const" only exists in STDC */
#else
#define CONST
#endif

/* Note: GOOD and BAD are already defined (differently) on RS6000 aix */
/* #define GOOD(status) ((status)&1)  VMS style status: test bit 0 */
/* #define BAD(status) (!GOOD(status)) Bit 0 set if OK, otherwise clear */

#ifndef BOOLEAN_DEFINED
typedef char BOOLEAN; /* Logical value */
#ifndef TRUE
#define TRUE (BOOLEAN)1
#endif
#define FALSE (BOOLEAN)0
#endif
#define BOOLEAN_DEFINED
#endif

#ifndef BOOL
#define BOOL BOOLEAN
#endif
#ifndef YES
#define YES (BOOLEAN)1
#define NO (BOOLEAN)0
#endif

#define TCP_PORT 2784 /* Arbitrary value -- should be allocated */

/* Is character c white space? */
#ifndef NOT ASCII
#define WHITE(c) (((unsigned char)(c))==' ') /* Assumes ASCII but faster */
#else
#define WHITE(c) (((c)==' ') || ((c)=='\t') || ((c)=='\n') || ((c)=='\r') )
#endif
The following names are undefined:
IBMLeague
/* THIS EXEC FILE COMPILES AND LINKS A C PROGRAM */
/* arg name */
"GLOBAL TXTLIB EDCBASE IBMLIB CMSLIB"
"CMOD HTDAEMON FINDGATE"
/* "EXEC TCPLOAD" name "1400CERN C" */
/*MOD" name */
address COMMAND

preflibs = 'NAPL HANDYPAK MK2 M2LIB BTM FORTSLAC VSF2FORT'
extralib = 'IBMLIB EDCBASE'
cdisk = 0

parse upper arg cname .
if cname = ' ' | cname = '?' then call ERR 1

/* is C370 disk up? */
'EXEC CMSSTACK LIFO QUERY DISK'
do queued()
   parse upper pull label vaddr mode rwstat .
   if label = 'IBM-C' then cdisk = 1
end
if cdisk = 0 then 'EXEC GIME C370'
'EXEC CC 'cname
if rc ^= 0 then call ERR 2
'GLOBAL TXTLIB EDCBASE IBMLIB'
'GLOBAL LOADLIB EDCLINK'
'EXEC CPLINK 'cname
if rc ^= 0 then call ERR 3
'EXEC CMOD' cname
if rc ^= 0 then call ERR 4
'GLOBAL TXTLIB 'preflibs extralib

exit
/* ------------------- EOProgram ------------------- */

ERR:
parse upper arg errno

select
   when errno = 1 then do
      'EXEC HELP CC'
   end
   when errno = 2 then do
      say 'Your C file failed to compile'
   end
   when errno = 3 then do
      say 'CPLINK failed'
   end
   when errno = 4 then do
      say 'CMOD failed'
   end
   otherwise nop
end /* of select */
exit
ADDRESS command /* PAD 3 before each PIPE STACK is a "fix" for the */
arg iinnpp /* problem with ?? being taken as EOF by IBM C */
Say ' NICFOUN1 Start->iinnpp
parse var iinnpp fid '((dispmode='/glued:'/versions'))
versions=translate(versions,' ','+')
dispmode=translate(dispmode,' ','+')
Queue 'SEEINDEX HREF='</FIND>'
If fid='1' Then do; queue 'No input whatsoever'; call get_out; end
parse var fid group '.disksname'.ftp'.fn
Say ' mode='dispmode 'glued='glued 'versions='versions
SELECT /*
WHEN datatype(dispmode,'X') THEN DO
  CALL GET_MODE group'.disksname 0;m_=result
  queue '<PLAINTEXT>'
  parse var dispmode _1 5 _2 7
    _1=x2d(_1)-1; _2=x2d(_2)
  'PIPE < 'fn ft m_1 | drop '1' | take '_2' | PAD 3 | STACK'
END /* dispmode is a hexadecimal number --> line numbers='X' */
WHEN dispmode='X' THEN DO
  Call GET_MODE group'.disksname 0;m_=result
  If glued='A' Then do
    pft='PRO:||substr(ft,5)
    CALL GET_MODE group'.P 'disksname 0;m_p=result
  End; Else do;m_p=m_;pft=ft;end
  'EXECIO * DISKR 'fn pft m_p' (STEM __. MAR 9 * FIND /cm EXP://
  If rc>0 Then,
    'EXECIO * DISKR 'fn pft m_p' (STEM __. MAR 9 * FIND /cm EXP://
      queue '<title>'._1'</title>'
      queue '<PLAINTEXT>'
    If ft='HELPROG' Then 'PIPE < 'fn ft m_,
        | NFIND . | SPEC 12-90 1 | PAD 3 | STACK'
  Else 'PIPE < 'fn ft m_ | NFIND . | PAD 3 | STACK'
END /* of dispmode='X' */
WHEN dispmode='W' THEN DO
  pft='PRO:||substr(ft,5)
  CALL GET_MODE group'.P 'disksname 0;m_p=result
  'EXECIO * DISKR 'fn pft m_p' (STEM __. MAR 9 * FIND /cm EXP://
  If rc>0 Then,
    'EXECIO * DISKR 'fn pft m_p' (STEM __. MAR 9 * FIND /cm EXP://
      queue '<title>'._1'</title>'
    'EXECIO * DISKR 'fn pft m_p' (STEM __. MAR 9 * FIND /cm TYP://
      If rc>0 Then types=''; else types=_1
      Select
        When find(versions,'H') >0 Then do
          queue '<PLAINTEXT>'
          Call GET_MODE group'.disksname 0;m_=result
          'PIPE < 'fn ft m_ | PAD 3 | STACK'
        End
        When find(versions,'S') >0 Then do
          parse var types 'SOURCE='sft .
          If sft=' ' Then do
            Call GET_MODE group'.S 'disksname 0;m_=result
            'PIPE < 'fn sft m_ | PAD 3 | STACK'
          End
          Else queue 'Something wrong with source'
        End
        Otherwise queue 'No displayable version available'
      End
    END /* of dispmode='W' */
WHEN dispmode='H' THEN DO
  Call GET_MODE group'.diskname 0;m_=result
  If glued='A' Then do
    pft='PRO:||substr(f1,5)
    CALL GET_MODE group'.p_.diskname 0;m_p=result
  End; Else do;m_p=m_p;pft=f1;end
  'EXECIO * DISKR fn pft m_p' (STEM _ MAR 9 * FIND /cm EXP:;'
    If rc>0 Then,
      'EXECIO * DISKR fn pft m_p' (STEM _ MAR 9 * FIND /CM EXP;'
      queue '<title>1</title>'
      queue '<PLAINTEXT>'
      If ft='HELPROG' Then 'PIPE <' fn ft m_p_,
        | NFIND . SPEC 12-90 1 | PAD 3 | STACK'
    Else 'PIPE <' fn ft m_p_ | NFIND . PAD 3 | STACK'
  END /* of dispmode='H' */
OTHERWISE queue 'Something is wrong, tell BERND@CERNVM'
END /*
/* Pipe Stack | > bla bla a | console */
CALL GET_OUT
RETURN queued()

*******************************************************************************/
GET_MODE: procedure; arg disk flag
'GLOBALV SELECT :XFIND 0 GET ACC_DISKS '
parse var acc_disks acc_disks ',' acc_links ',' acc_modes
nn_n=find(acc_disks,disk)
if nn_n>0 Then RETURN subword(acc_modes,nn_n,1)
'GLOBALV SELECT :XFIND 1 GET 'disk'
parse value value(disk) with id addr uaddr umode rpw .
if id='' then do
  Queue 'The disk 'disk' is not defined'
  Call Get_out
end
If length(umode)=1 Then flag=1 /* disk accessed for all modes */
If flag & umode ^='' & umode ^='' then RETURN umode
else do
  l=qdisk('?','ADDRESS')
  'EXECIO 1 CP (SKIP STRING LINK TO 'id addr 'AS' l_ 'RR' rpw'
  If rc>2 Then Do
    Queue 'Linking problem with disk 'disk': TELL BERND@CERNVM'
    Call Get_out
End;
zz=typemode('HT');m_=QDISK('?','MODE','-Z'); 'ACCESS 'l_ m_;zz=typemode(zz)
If rc>0 Then Do
  Queue 'Accessing problem with disk 'disk': TELL BERND@CERNVM'
  Call Get_out
End
acc_disks=acc_disks' 'disk
acc_links=acc_links' l_
acc_modes=acc_modes' 'm_
'GLOBALV SELECT :XFIND 0 SETL ACC_DISKS 'acc_disks','acc_links','acc_modes
end
RETURN m
*******************************************************************************/
GET_OUT:
'GLOBALV SELECT :XFIND 0 GET ACC_DISKS '
parse var acc_disks acc_disks ',' acc_links ',' acc_modes
'SET CMSTYPE HT'
do k-1 to words(acc_links)
'RELEASE 'WORD(acc_modes,k)
'EXECIO 1 CP (SKIP STRING DETACH 'WORD(acc_links,k)
end
'SET CMSTYPE RT'
'GLOBALV SELECT :XFIND_0 PURGE'
Say ' NICFOUN1 End -> stacksize='queued()
EXIT queued()
-compile
htdaemon c
httcp c
findgate c

-CC
include ebdlib
cc name "(def ....)"

-header files
HTUtils.h
tcp.h
HTTCP.h

cw
hime watchnew
cw fn.c (prm debug t prm vm t

tcp/ip
gime tcpappl
files: manifest h

c370
gime c370
cc fn
help file: cc helpcms
cc fn (xref list
cc fn (langlvl (ansi)

duplication
string h          comment out strncpy
edcbase txtlib   no different from IBM-C'S version

-error with HTDaemon c
bsdtypes h       (time t was commented out in bsdtypes.h, put it in)
strncpy           comment out strncpy from string.h
logfile           change logfile to logfiles in HTDaemon.c
                          real fix: change LogFile to logfile

-error with HTTCP c
sin_addr          HTDaemon (structure sockaddr_in defined by in.h by tcp.h
fix               cc fn (langlvl (ansi) in cc2 exec
                          real fix: find newer version, WWW/Imple../HTTCP.c
port/index        fix: define index in string.h
                          real fix: use strchr() instead of index

-vm tools
disklist          who has what
qdiskid           sho user addr
q txtlib           query current global txtlib
txtlib map fn generate map for a txtlib
/* REXX please  - Run Daemon */
address command
Say 'Access the disks for TCP/IP '
'EXEC GIME MAINT 5E5 (MODE -P' */
'EXEC GIME TCPMNT 2B0 * READIT (MODE -P' /* TCP/IP source */
'EXEC GIME TCPMNT 410 * DACUR (MODE -P' /* TCP/IP base */
'EXEC GIME TCPMNT 412 * DACUR (MODE -P' /* TCP/IP text */
'EXEC GIME TCPMNT 441 * DACUR (MODE -P' /* TCP/IP last updates fixes */
Say 'Indicate to CMS which libraries to search'
'GLOBAL TXTLIB EDCBASE IBMLIB CMSLIB'
Say 'Access the disk with all the indices of XFIND'
'EXEC DROP Z'
'EXEC GIME TESTSERV 201 Z'
Say 'Start the DAEMON'
'HTDAEMON -v -a *:8000 -l findlog'
```c
/**
   'EXEC GLOBADD LOADLIB EDCLINK'
   '*htdaemon -v -a *:2784' */
   'htdaemon -v -a *:2784'
```
/ * SET UP TCP ACCESS */
/ * "GIME TCPMNT 582" tyh */
"GIME TCPAPPL"
"GLOBAL TXTLIB IBMLIB CMSLIB EDCBASE COMMTXT VSPASCAL AMPLANG"
RECOVER

Searching ...

*** Note: SPICELL is not now a valid VM userid ***

There are 16 copies of SPICELL 192 on backup tapes. They are:

1 ) SPICELL 0192 dumped to file 257 of tape RL2748 on 10/03/95 at 19:45:10
2 ) SPICELL 0192 dumped to file 214 of tape RL1816 on 07/04/95 at 20:04:19
3 ) SPICELL 0192 dumped to file 81 of tape RL1698 on 04/04/95 at 22:01:55
4 ) SPICELL 0192 dumped to file 216 of tape RL2993 on 01/03/95 at 22:30:20
5 ) SPICELL 0192 dumped to file 136 of tape RL3154 on 10/04/94 at 20:37:49
6 ) SPICELL 0192 dumped to file 248 of tape RL0457 on 07/06/94 at 20:51:40
7 ) SPICELL 0192 dumped to file 168 of tape RL3326 on 04/06/94 at 03:28:17
8 ) SPICELL 0192 dumped to file 185 of tape RL3139 on 01/06/94 at 04:25:52
9 ) SPICELL 0192 dumped to file 27 of tape RL2980 on 10/06/93 at 03:06:46
10) SPICELL 0192 dumped to file 10 of tape RL2741 on 07/07/93 at 04:12:40
11) SPICELL 0192 dumped to file 136 of tape RL2217 on 04/07/93 at 01:42:14
12) SPICELL 0192 dumped to file 139 of tape RL2544 on 01/06/93 at 01:32:56
13) SPICELL 0192 dumped to file 176 of tape RL2060 on 10/07/92 at 09:11:56
14) SPICELL 0192 dumped to file 170 of tape RL1559 on 07/08/92 at 04:14:00
15) SPICELL 0192 dumped to file 30 of tape RL1798 on 04/08/92 at 02:39:53
16) SPICELL 0192 dumped to file 51 of tape RL1414 on 01/08/92 at 01:02:19

Which one of the above do you wish to restore?
(1 - 16 , <CR>=none)
SLACVM SPIRES HEP Preprint Database

Search
   Perform search using standard SPIRES terms.
Help
   Get help for SPIRES
You can search this index. Type the keyword(s) you want to search for:

SLAC SPIRES HEP Preprint database search

Use standard SPIRES search terms such as...

find author Perl, M
find title tau and date 1980
STORES: Stores Online Catalog Search Facility

Use the STORES command to search the Online Stores Catalog for
catalog numbers, descriptions, prices, units of issue, and quantity
balances (SLAC EXEC). This command does not replace existing
ordering procedures.

Syntax:

[function] value

Functions:

<table>
<thead>
<tr>
<th>ALLDes</th>
<th>allows any search word(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAMDes</td>
<td>requires Family Description search word(s)</td>
</tr>
<tr>
<td>ITEMDes</td>
<td>requires Item Description search word(s)</td>
</tr>
<tr>
<td>ALLNum</td>
<td>requires Catalog Identifier number</td>
</tr>
<tr>
<td>FAMNum</td>
<td>requires Catalog Identifier number</td>
</tr>
<tr>
<td>ITEMNum</td>
<td>requires Catalog Identifier number</td>
</tr>
<tr>
<td>SPCode</td>
<td>requires Spare Parts Code</td>
</tr>
</tbody>
</table>

PARAM:

function determines what portion of the Catalog will be searched (family, stock
item or both), and how the Catalog will be searched (by Description or
Catalog Identifier number). The following are valid functions:

ALLDes displays both the family description and stock item catalog numbers
that match the description search word(s) given in "value." Unit of
issue, issue price, and balance on hand, are included for stock items.
EXAMPLE: ALLDES OIL CUTTING

FAMDes displays all family catalog numbers and their descriptions that match
the family description search word(s) given in "value."
This function provides a display similar in content to the printed
Stores Catalog index.
EXAMPLE: FAMDES PIPE

ITEMDes displays all stock item catalog numbers that match the item descrip-
tion search word(s) given in "value." The output also includes unit
of issue, issue price, and balance on hand.
EXAMPLE: ITEMDES PENCIL BLACK | PENCIL BLK

ALLNum displays both family description and stock item catalog numbers that
match the catalog identifier number given in "value." Unit of issue,
issue price, and balance on hand, are included for stock items.
EXAMPLE: ALLNUM 68

FAMNum displays all family catalog numbers and their descriptions that match
the catalog identifier number given in "value."
EXAMPLE: FAMNUM 55

ITEMNum displays all stock item numbers that match the catalog identifier
number given in "value." The output also includes unit of issue,
issue price, and balance on hand.
EXAMPLE: ITEMNUM 684

SPCode displays both family description and stock item catalog numbers that match the spare parts code given in "value." Unit of issue, issue price, and balance on hand, are included for stock items.

EXAMPLE: SPCODE A099

value is the search term and may contain a catalog description search word(s), optionally combined by search qualifiers, a catalog identifier number, or a spare parts code, depending on which function is specified.

SEARCH WORD(s): (For ALLDes, FAMDes, and ITEMDes Functions)

Search words are words contained in either the family description or stock item description, such as PENCIL BLACK MECH. They must be separated by blank spaces and may be combined with search qualifiers.

See Note 2 for important considerations in utilizing the Stores Online Catalog.

Search qualifiers are used to limit or expand a search. A blank space between two search words represents a logical AND, for inclusion purposes. The most useful qualifiers are as follows:

```
.fo 
Search Qualifier  Abbreviation  Example of Value
AND             &  PENCIL BLACK & PENCIL MECH
OR               |  PENCIL BLACK | PENCIL BLUE
STRING          |  STRING OLD would find all descriptions
               with words containing the string
               OLD, such has HOLD, HOLDER, etc.
```

Truncated searches may be accomplished by suffixing a star "*" to search word(s). In the following example, all words beginning with PEN BL and ME would be found.

EXAMPLE: PEN* BL* ME*

Also, compound searches can be done.

EXAMPLE: PENCIL RED | PENCIL BLUE & PENCIL MECH

See Note 4 for further information concerning search qualifiers.

CATALOG IDENTIFIER NUMBER:

For ALLNum and FAMNum Functions:

A catalog number contains either 10 characters (general stores, metal stores) or 14 characters (spare parts). See Note 1, below.

A catalog identifier number is the stem of a catalog number, from 2 to 8 digits which may represent:

GROUP (2 digits), such as 55, or
CLASS (5 digits), such as 55200, or
FAMILY (8 digits), such as 55200600.

Group 59 is too large a class to display and may therefore not be specified as a catalog identifier number.

Three or more digits may also be entered to further slice groupings, such as 553, or 552006.

Catalog Identifier Numbers may be entered with or without dashes or blanks separating Group, Class, Family, etc., such as 55-300-020, 55 300 020, 55300020.

For ITEMNum Function:

In addition to the above, a complete catalog number, up to 14 digits, may be given, such as 5323000105, or 5930120001A002.

SPARE PARTS CODE:

The Spare Parts Code is an owner group identification code for spare parts maintained by SLAC Stores. This code may be used only with the SPCode function, and consists of 4 characters, such as A048, A099. Optionally, the first character, "A" may be omitted.

NOTES:

Note 1: GENERAL DESCRIPTION OF THE STORES CATALOG

The Stores Catalog contains descriptions of over 9000 stock items, including general stores items, metal stores items, and spare parts. All of these items are arranged or subdivided by group, class, family, and item, as follows:

55-300-020-06 LUMBER, FIRE-RETARDANT, CONST.GRADE S4S, 2X4 CONST.FIR

Description
---------------------------------
ITEM: 2X4 CONST.FIR, FIRE RETARDANT
FAMILY: LUMBER, FIRE-RETARDANT, CONST.GRADE S4S
CLASS: [Fire Retardant Lumber and Plywood]
GROUP: [Lumber and Building Materials]

Thus, in the example above, the catalog group would be 55, the catalog class would be 55-300, the catalog family would be 55-300-020, and the catalog item or catalog stock number would be 55-300-020-06.

For spare parts, a 4 character suffix is included for both family and catalog items, such as 31-300-001-08A099.

Searchable descriptions exist at the FAMILY and ITEM levels. Notice from the above example that Lumber appears in the FAMILY description but not in the ITEM description, and that 2X4 and Fir appear only in the ITEM description and not the FAMILY description. Using ALLDes, FAMDes, or ITEMDes you can search both descriptions, or only family descriptions, or only item descriptions.
Once you know the group, class, or family identifier numbers, you can then use FAMNum to display all family descriptions within a class or group. Use ITEMNum to display all items within a family or class. Or, you can use ALLNum to display both family descriptions and items within a family or class.

Note from the above that using more than two specific words in FAMDes may prove to be misleading because most family descriptions are general rather than specific. Using ALLDes or ITEMDes (rather than FAMDes) will yield more satisfactory results for very specific word searches, such as ALLDES PENCIL BLACK MECH THIN.

It is important to be aware of the fact that description words and catalog identifier numbers cannot be mixed in any of the Stores functions.

In using search word(s) it is important to note that for most stock items, the descriptive word(s) are singular rather than plural. Searching on PENS or PENCILS will return a message that no descriptions were found. Searching on PEN or PENCIL would return many descriptions. Searching on COLORS, however, would return several descriptions, mostly those containing the word ASSORTED.

See Note 2 for significant considerations in using the Stores Online Catalog.

Occasionally during FAMDes or FAMNum displays you may notice two identical family identifier numbers with similar or identical descriptions. This is due to spare parts family identifiers additionally containing a 5 character spare parts owner code, which is not displayed during FAMDes or FAMNum displays. Executing a ALLDes or ALLNum display for the same search term will show you both numbers. For the sake of displaying the maximum length of description possible in the FAMDes and FAMNum displays, the '00' suffix (which would otherwise be Item above if it were not a family description) as well as any spare parts code, has been dropped. In the ALLDes and ALLNum displays a non spare parts family description also has the '00' suffix dropped to further distinguish family descriptions from item descriptions, but spare parts family description identifier numbers are completely displayed.

To see both family and item descriptions for a particular spare parts code, use the SPCode function, such as STORES SPC A099.

Suggestions and comments concerning the content of the Stores catalog may be directed to the following individuals, in the order given:
Lillie Copeland, X4253; Thomas Butler, X3759; Bob Ratko, X4107;
Jerry Belk, X2484.

Note 4: SEARCH QUALIFIERS

Stores search qualifiers are relational operators identical to those used in the SPIRES database language for searching indexes. With the exception of STRING, PREFIX, and SUFFIX, most may be of little use for searching the Stores Catalog description field because of the way in which the descriptions are indexed. All descriptions are broken into words at every occurrence of (',',:'/'), etc). Also, no integer values appear in any description, only string or character values. Thus, equality, inequality, word, with, and having operators may be in many instances of little or no value. Because of the composition of the Stores
Function statement, no parentheses are allowed within Value. For more information concerning SPIRES index searching, see Sections B.3.2.3 and B.3.3.1 of the SPIRES Searching and Updating Manual.

EXAMPLES:

1. For many items, the default function ALLDes will suffice to find your goal. To find 1/2" magic mending tape, for instance,
   Enter: MAGIC MENDING TAPE
   This would show you that 2 sizes are available, including your goal.

2. However, due to the way family and item descriptions are frequently constructed, it may take several commands to find your goal. Suppose you were interested in finding out the availability of aluminum electrolytic capacitors with WVDC of 15 rated at 21000 MFD. One satisfactory approach would be to do the following:
   Enter: STOReS CAPACITOR ALUMINUm ELECTROLYTIC
   This reveals that there are 7 families. To save the time it would take to look through all of those families and potentially pick up all sorts of non-interest items, one might instead try:
   Enter: WVDC 15 21000 MFD
   This yields one item which may or may not be a capacitor of aluminum construction within the electrolytic category, so, to check that possibility we
   Enter: ALLNum 59312909
   which verifies that the previously displayed item was, in fact, a capacitor aluminum electrolytic WVDC 15 2100 MFD.
SLAC STORES Catalog and item listing

Find
Enter an item or partial description

Help
Help with STORES
You can search this index. Type the keyword(s) you want to search for:

Search the SLAC Stores Catalog.

Enter an item or portion of item description

   pipes
   paper
   <END>
#ifndef __string
#define __string

/***************************************************************************/
/* <string.H> header file */
/* IBM C/370 Library */
/* Program Number: 5688-039 */
/* Version: 1.0 */
/* Release: 2.0 */
/* Modification: 0.0 */
/* Restricted Materials of IBM */
/* */
/* Copyright: */
/* International Business Machines Corporation 1988, 1989 */
/***************************************************************************/

#ifndef __size_t
    typedef unsigned int size_t;
#define __size_t 1
#endif
#endif

void * memchr (const void *, int, size_t);
#define memcpyn __memcpy
#pragma linkage (__memcpy,builtin)
void * (__memcpy) (void *, const void *, size_t);
#define memcmpn __memcmp
#pragma linkage (__memcmp,builtin)
int (__memcmp) (const void *, const void *, size_t);
#define memsetn __memset
#pragma linkage (__memset,builtin)
void * (__memset) (void *, int, size_t);
#define memcmpn __memcmp
#pragma linkage (__memcmp,builtin)
int (__memcmp) (const void *, const void *, size_t);
void * (memset) (void *, int, size_t);
void * (memcpy) (void *, const void *, size_t);
int (strcmp) (const void *, const void *);
char * strchr (const char *, int);
int strcmp (const char *, const char *);
int strlen (const char *);
#define strpyn __strpyn
#pragma linkage (__strpyn,builtin)
char * (__strpyn) (char *, const char *);
char * (strcmp) (char *, const char *);
#define strc aging(n) __strc aging(n)
#pragma linkage (__strc aging,n,builtin)
char * (__strc aging,n) (char *, const char *);
char * (strncpy) (char *, const char *, size_t); terryh*/
size_t strcspn (const char *, const char *);
char * strerror (int);
#define strlenn __strlenn
#pragma linkage (__strlenn,builtin)
size_t (__strlenn) (const char *);
size_t (strlen) (const char *);

8/29/01
#define strcmp(x,y) __strcmp(x,y)
#pragma linkage(__strcmp, builtin)
int __strcmp (const char *, const char *);
int  strcmp (const char *, const char *);
char * strncat (char *, const char *, size_t);
int  strncmp (const char *, const char *,
               size_t);
char * strchr (const char *, int);
size_t strspn (const char *, const char *);
char * strstr (const char *, const char *);
char * strtok (char *, const char *);
size_t strxfrm (char *, const char *, size_t);
*/ extern char *index();  * terryh */
#endif
System-system differences for TCP include files and macros

This file includes for each system, the files necessary for network and file I/O

History:
22 Feb 91 Written (TBL) as part of the WWW project.

/*
#define NETCLOSE close  /* Routine to close a TCP-IP socket */
#define NETREAD read    /* Routine to read from a TCP-IP socket */
#define NETWRITE write   /* Routine to write to a TCP-IP socket */

/* On the NeXT, there's a little package of include files. */
#ifdef NeXT
#include <libc.h>
#define ntohs(x) (x)   /* NeXT has all this packaged up */
#define htonl(x) (x)
#include <sys/errno.h>
#define SELECT        /* Get ECONNRESET etc */
#define INCLUDES_DONE /* Is supported ok */
#else
#include <stdio.h>

/* MVS is compiled as for VM. MVS has no unix-style I/O
   The command line compile options seem to come across in
   lower case.
   See also lots of VM stuff lower down.
*/
#endif mvs
#define MVS
#endif MVS
#define VM
#endif

/* VM doesn't have a built-in predefined token, so we cheat: */
#ifdef _STDIO_
#define VM
#else
#include <string.h>   /* For bzero etc - not NeXT or VM */
#define SELECT        /* Handle >1 channel if we can. */
#endif
#endif

/* Under VMS, there are many versions of TCP-IP. Define one if you
don't use Digital's UCX product:

   UCX     DEC's "Ulrix connection" (default)
   WIN_TCP From Wollongong, now GEC software.
   MULTINET From SRL, now from TGV Inv.

The second two do not interfere with the unix i/o library, and so they

need special calls to read, write and close sockets. In these cases the
socket number is a VMS channel number, so we make the HORRIBLE
assumption that a channel number will be greater than 10 but a
unix file descriptor less than 10.

+/
#endif
#endif WIN_TCP
#endif NETREAD
#endif NETWRITE
#endif NETCLOSE
#define NETREAD(s,b,l) (((s)>10 ? netread((s),(b),(l)) : read((s),(b),(l))))
#define NETWRITE(s,b,l) (((s)>10 ? netwrite((s),(b),(l)) : write((s),(b),(l))))
#define NETCLOSE(s) (((s)>10 ? netclose(s) : close(s))
#endif MULTINET
#endif NETCLOSE
#endif NETREAD
#endif NETWRITE
#define NETREAD(s,b,l) (((s)>10 ? socket_read((s),(b),(l)) : read((s),(b),(l))))
#define NETWRITE(s,b,l) (((s)>10 ? socket_write((s),(b),(l)) : 
                      write((s),(b),(l))))
#define NETCLOSE(s) (((s)>10 ? socket_close(s) : close(s))
#endif

/* Certainly this works for UCX:  @@@
+/
#include types
#include errno
#include time
#include string
#include stdio
#include file
#include unixio
#define INCLUDES_DONE

#include socket
#include in
#include inet
#include netdb
#define TCP_INCLUDES_DONE
#endif  /* vms */

/* IBM VM/CMS or MVS
 ** ---------------------
 **
 ** Note: All files must have lines <= 80 characters
 **
 ** Under VM, compile with "DEF=VM"
 **
 ** Under MVS, compile with "NOMAR DEF(MVS)" to get rid of 72 char margin
 ** System include files TCPIP and COMMMAAC need line number removal(!)
 */
#endif VM
#define NOT_ASCII /* or MVS -- see above. */
#define NO_UNIX_IO /* char type is not ASCII */
/* Unix I/O routines are not supported */
#define SHORT_NAMES
#include <manifest.h>
#include <bsdtypes.h>
#include <stdbool.h>
#include <socket.h>
#include <in.h>
#include <netdb.h>
#include <errno.h>
extern char asciitoebcdic[], ebdictoascii[];
#define TOASCII(c) (c==\'\n' ? 10 : ebdictoascii[c])
#define FROMASCII(c) (c== 10 ? \'\n' : asciitoebcdic[c])
#include <sys/types.h>
#include <string.h>
#define INCLUDES_DONE
#define TCP_INCLUDES_DONE
#endif

/* Regular BSD unix versions: */
#ifndef INCLUDES_DONE
#include <sys/types.h>
/* include <streams/stream.h> not ultrix */
#include <string.h>
#include <stdio.h>
#include <errno.h>
/* independent */
#include <sys/time.h>
/* independent */
#include <sys/stat.h>
#include <sys/param.h>
#include <sys/file.h> /* For open() etc */
#endif /* Normal includes */

/*
 Default include files for TCP
*/
#ifndef TCP_INCLUDES_DONE
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h> /* Must be after netinet/in.h */
#include <netdb.h>
#endif /* TCP includes */

/*
 Default macros for manipulating masks for select()
*/
#ifndef FD_SET
typedef unsigned int fd_set;
#define FD_SET(fd,pmask) (*pmask) |= (1<<fd)
#define FD_CLR(fd,pmask) (*pmask) &=(1<<fd)
#define FD_ZERO(pmask) (*pmask)=0
#define FD_ISSET(fd,pmask) (*pmask) & (1<<fd))
#endif

/*
 Default macros for converting characters
*/
#ifndef TOASCII
#define TOASCII(c) (c)
#define FROMASCII(c) (c)
#endif
-----+ Execute "RSCOV193"

** RECOVER TERRYH 193 to TDISK **
DASD 0555 DEFINED
TAPE 0F13 ATTACHED TO BATCH07A 0181
TAPE 0181 ON DEV 0F13 R/W
MNTREQ1001 18:40:53 Mounting volume RL0357 on device 181, SL NoRing
MNTVER1011 18:41:18 Volume RL0357 on device 181 has been verified
Rewind complete
RESTORING TYYH193
DATA DUMPED 10/31/91 AT 04.35.56 GMT FROM TYYH193 RESTORED TO SCRATCH
INPUT CYLINDER EXTENTS  OUTPUT CYLINDER EXTENTS
 START STOP START STOP
 0000 0000 0000 0000

END OF RESTORE

END OF JOB
18:42:59 * MSG FROM VMSETUPA: VMSCAN0011 VMSETUP REQUEST FOR TERRYH TYYH239 WILL BE
TAPE 0181 DETACHED BY VMSETUPA, TAPE VOLUME(S) UNLOADED

DMSCAR002E FILE(S) NOT FOUND.

FILENAME   FILETYPE   FM FORMAT LRECL   RECS   BLOCKS   DATE   TIME   LABEL
 A          ERRS       Z1 V           49    1        10/16/91 18:13:27 TYYH193
 A          TEXT       Z1 F           80    5        10/16/91 18:15:35 TYYH193
 BSDTYPES   H          Z1 V           70   120      10/15/91 18:01:30 TYYH193
 CC_ALL     EXEC       Z1 V           81   13       10/14/91 15:29:29 TYYH193
 CC2        EXEC       Z1 V           58   8        10/30/91 14:18:05 TYYH193
 CMOD       EXEC       Z2 V           62   201      10/30/91 15:38:28 TYYH193
 CMS        EXEC       Z1 F           80   13       10/19/91 10:50:58 TYYH193
 CMS        PRESERVE  Z1 V           65   30       10/30/91 16:15:36 TYYH193
 CPOBJ      RMAP       Z1 F           80   3        10/16/91 18:45:59 TYYH193
 DLOAD      EXEC       Z1 V           62   6        10/30/91 15:35:54 TYYH193
 EDCXV      C          Z1 F           80   5        10/16/91 18:08:43 TYYH193
 FGET       EXEC       Z2 V           78   129      2/9/26/91 11:30:24 TYYH193
 FINDGATE   C          Z1 V           78   245      2/9/26/91 9:58:03 TYYH193
 FINDGATE   C2        Z2 V           78   245      2/10/91 15:28:41 TYYH193
 FINDGATE   TEXT      Z2 F           80   113      3/10/91 14:18:46 TYYH193
 FSEARCH    EXEC       Z1 V           76   120      2/9/26/91 10:43:32 TYYH193
 HTDAEMON   C          Z1 V           85   639      5/10/15/91 18:10:19 TYYH193
 HTDAEMON   C2        Z1 V           85   639      5/10/15/91 15:45:38 TYYH193
 HTDAEMON   MODULE    Z1 V           65535   5      32/10/30/91 16:15:39 TYYH193
 HTDAEMON   TEXT     Z1 F           80   183      4/10/30/91 14:18:19 TYYH193
 HTTCP      C          Z1 V           79   282      2/10/91 17:04:54 TYYH193
 HTTCP      H          Z1 V           78   94       10/14/91 15:47:38 TYYH193
 HTTCP      TEXT     Z1 F           80   77       2/10/91 14:18:32 TYYH193
 HTUTLIS    H          Z1 V           79   85       10/10/91 15:47:21 TYYH193
 LOAD       MAP       Z1 F           100  392      10/30/91 15:34:23 TYYH193
 MAKE       EXEC      Z1 V           51   6        1/9/19/91 14:06:07 TYYH193
 MYCC       EXEC      Z1 V           63   51       10/16/91 18:28:29 TYYH193
 NICFONI    EXEC      Z2 V           78   125      2/9/24/91 11:01:53 TYYH193
 PROFILE    EXEC      Z1 V           72   28       1/9/26/91 10:44:28 TYYH193
Q      Z1 F      80      52
RUN    EXEC     Z1 V      74      21
SETUP  EXEC     Z1 V      62      4
STRING H      Z2 F      80      75
TCP    H      Z1 V      91     180
-----> Batch return code = 0.

-----> System cleanup $BCLEAN.

*****************************************************************************
* End execution of TYH239 for TERRYH at 18:42:59 on 14/11/91. *
* Actual CPU time (mm:ss[.th]): Virtual = 0:00.37, Total = 0:00.64 *
* I/O counts: SIO-000531 RDR-000000 PRT-000100 PCH-000071 *
* Space used on A-disk at job end: 48K *
* I/O counts on units still available *
* Spool units *
* 0009: 87 000C: 0 000D: 2 000E: 0 *
* 00AE: 0 *
* Dasd units *
* 0190(S): 25 0191(A): 76 019E(Y): 45 0555(Z): 22 *
* 059E(T): 15 059F(U): 5 05A0(V): 32 *

*****************************************************************************
UNIX Command Summary with an Extract of Output

Commands used to copy files RECOVERed from CMS Backup to WINTERS 294
(a SPACE TEMPDISK), from that SLACVM minidisk on to UNIX AFS WWW archive
space:


The commands (almost always) and output were cut and pasted from the
xterm (or aixterm) window where the work was executed. Text in square brackets
([]) is commentary.

pwd

date
Wed Jul 22 03:15:11 PDT 1998

vmnfs refresh winters.294
[SLACVM.SLAC.STANFORD.EDU]
Disk WINTERS.294 refreshed.

cp -p /nfs/slacvm/ascii/winters.294/* .
762 cassandra /192/r11559> chmod a-x *
763 cassandra /192/r11559> mv readme.r11559 README
764 cassandra /192/r11559> mv spice192.backup spicell192.backup

chmod u+w README

ls -1 |grep rw
-rw-r--r-- 1 winters sf 10787 Jul 22 03:37 README
-rw-r--r-- 1 winters sf 560 Jul 22 03:15 unix-commands

ls -l winters.filelist
-r---r--- 1 winters sf 7236 Jul 21 20:33 winters.filelist
771 cassandra /192/r11559> ls -l >winters.ls

ls -1 >winters.ls
772 cassandra /192/r11559> chmod a-x winters.ls
773 cassandra /192/r11559> chmod u-w winters.ls
774 cassandra /192/r11559> ls -l winters.ls
-r---r--- 1 winters sf 4598 Jul 22 03:39 winters.ls

wc winters.ls
71 632 4598 winters.ls
776 cassandra /192/r11559> wc winters.filelist
67 871 7236 winters.filelist

grep -i filelist winters.filelist
778 cassandra /192/r11559> grep -i filelist winters.ls
-r---r--- 1 winters sf 7236 Jul 21 20:33 winters.filelist
779 cassandra /192/r11559> grep -i winters.ls winters.ls
-rw-r---r-- 1 winters sf 0 Jul 22 03:39 winters.ls
780 cassandra /192/r11559> grep -i total winters.ls
780 cassandra /192/r11559> grep -i total winters.ls
781 cassandra /192/r11559> grep -i unix-commands winters.ls
-rw-r---r-- 1 winters sf 560 Jul 22 03:15 unix-commands
[71 = 67 + winters.filelist + winters.ls + "total" + unix-commands]

date

Dates on unix-commands and README are likely to be newer than the dates for those files in winters.ls.

22 Jul 1998

Joan M. Winters
<table>
<thead>
<tr>
<th>Program</th>
<th>Type</th>
<th>File</th>
<th>Size</th>
<th>Time</th>
<th>Date/Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>H</td>
<td>A1 V</td>
<td>91</td>
<td>180</td>
<td>2 10/14/91 15:47:28 SPC2</td>
</tr>
<tr>
<td>HTUTILS</td>
<td>H</td>
<td>A1 V</td>
<td>79</td>
<td>85</td>
<td>1 10/14/91 15:47:21 SPC2</td>
</tr>
<tr>
<td>CC.ALL</td>
<td>EXEC</td>
<td>A1 V</td>
<td>81</td>
<td>13</td>
<td>1 10/14/91 15:29:29 SPC2</td>
</tr>
<tr>
<td>FINDGATE</td>
<td>C2</td>
<td>A1 V</td>
<td>78</td>
<td>245</td>
<td>2 10/14/91 15:28:41 SPC2</td>
</tr>
<tr>
<td>NICFOUN1</td>
<td>EXEC</td>
<td>A2 V</td>
<td>78</td>
<td>125</td>
<td>2 10/14/91 15:28:41 SPC2</td>
</tr>
<tr>
<td>RUN</td>
<td>EXEC</td>
<td>A1 V</td>
<td>74</td>
<td>21</td>
<td>2 9/24/91 11:01:53 SPC2</td>
</tr>
<tr>
<td>MAKE</td>
<td>EXEC</td>
<td>A1 V</td>
<td>51</td>
<td>6</td>
<td>1 9/19/91 14:19:57 SPC2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 9/19/91 14:06:07 SPC2</td>
</tr>
</tbody>
</table>
* Start execution of JCW122 for WINTERS, class S.
* Execution begins at 03:12:04 on 22 Jul 1998 in BATCH08A at SLACVM.
* Submitted at 03:11:59 on 22 Jul 1998 for WINTERS at SLACVM.
* Running CURCMS, mode XA, storage = 4096K, time limit = 2 minutes.
* Max print = 10K lines, max punch = 100K cards.

----> Execute "R$COV192"

** RECOVER SPICELL 192 to TDISK **
DASD 0555 DEFINED
TAPE OF13 ATTACHED TO BATCH08A 0181
TAPE 0181 ON DEV OF13 3490 R/W SUBCH = 000D
MNTREQ1001 03:12:05 Mounting volume RL1559 on device 181, SL NoRing
MNTVER1071 03:13:13 Remounting volume RL1559 on device 181; intervention required
MNTVER1011 03:13:41 Volume RL1559 on device 181 has been verified
Rewind complete
RESTORING WWW
DATA DUMPED 07/08/92 AT 04.13.53 GMT FROM WWW RESTORED TO SCRATCH
INPUT CYLINDER EXTENTS OUTPUT CYLINDER EXTENTS
  START  STOP  START  STOP
  0000  0004  0000  0004
END OF RESTORE
END OF JOB
03:15:06 * MSG FROM VMSETUP: VMSCAN0011 VMSETUP REQUEST FOR WINTERS JCW122 WILL BE TAPE 0181 DETACHED BY VMSETUP
CARD DUMP A C Z1
PUN FILE 0738 SENT TO WINTERS RDR AS 7405 RECS 0003 CPY 001 Y NOHOLD NOKEEP
CARD DUMP A ERS Z1
PUN FILE 0739 SENT TO WINTERS RDR AS 7406 RECS 0003 CPY 001 Y NOHOLD NOKEEP
CARD DUMP A EXEC Z1
PUN FILE 0740 SENT TO WINTERS RDR AS 7407 RECS 0005 CPY 001 Y NOHOLD NOKEEP
CARD DUMP A TEXT Z1
PUN FILE 0741 SENT TO WINTERS RDR AS 7408 RECS 0005 CPY 001 Y NOHOLD NOKEEP
CARD DUMP BINLIST @HTML Z0
PUN FILE 0742 SENT TO WINTERS RDR AS 7409 RECS 0006 CPY 001 Y NOHOLD NOKEEP
CARD DUMP BINLIST @INDEX Z0
PUN FILE 0743 SENT TO WINTERS RDR AS 7410 RECS 0004 CPY 001 Y NOHOLD NOKEEP
CARD DUMP BINLIST HELP Z1
PUN FILE 0744 SENT TO WINTERS RDR AS 7411 RECS 0004 CPY 001 Y NOHOLD NOKEEP
CARD DUMP BINLIST HTML Z1
PUN FILE 0745 SENT TO WINTERS RDR AS 7412 RECS 0006 CPY 001 Y NOHOLD NOKEEP
CARD DUMP BINLIST INDEX Z1
PUN FILE 0746 SENT TO WINTERS RDR AS 7413 RECS 0004 CPY 001 Y NOHOLD NOKEEP
CARD DUMP BINLIST SAVINDEX Z1
PUN FILE 0747 SENT TO WINTERS RDR AS 7414 RECS 0004 CPY 001 Y NOHOLD NOKEEP
CARD DUMP BSDTYPES H Z1
PUN FILE 0748 SENT TO WINTERS RDR AS 7415 RECS 0044 CPY 001 Y NOHOLD NOKEEP
CARD DUMP CC ALL EXEC Z1
PUN FILE 0749 SENT TO WINTERS RDR AS 7416 RECS 0005 CPY 001 Y NOHOLD NOKEEP
CARD DUMP CC2 EXEC Z1
PUN FILE 0750 SENT TO WINTERS RDR AS 7417 RECS 0006 CPY 001 Y NOHOLD NOKEEP
CARD DUMP CMOD EXEC Z2
PUN FILE 0751 SENT TO WINTERS RDR AS 7418 RECS 0057 CPY 001 Y NOHOLD NOKEEP
CARD DUMP CMS PRESERVE Z1
PUN FILE 0752 SENT TO WINTERS RDR AS 7419 RECS 0014 CPY 001 Y NOHOLD NOKEEP
CARD DUMP CPOBJ RMAP Z1

8/29/01
<table>
<thead>
<tr>
<th>FILENAME</th>
<th>FILETYPE</th>
<th>FM</th>
<th>FORMAT</th>
<th>LRECL</th>
<th>RECS</th>
<th>BLOCKS</th>
<th>DATE</th>
<th>TIME</th>
<th>LABEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILE name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>C</td>
<td>Z1</td>
<td>F</td>
<td>80</td>
<td>5</td>
<td></td>
<td>10/16/91</td>
<td>18:13:27</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>ERRS</td>
<td>Z1</td>
<td>V</td>
<td>49</td>
<td>1</td>
<td></td>
<td>10/16/91</td>
<td>18:15:35</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>EXEC</td>
<td>Z1</td>
<td>V</td>
<td>68</td>
<td>10</td>
<td></td>
<td>11/01/91</td>
<td>15:13:32</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>TEXT</td>
<td>Z1</td>
<td>F</td>
<td>80</td>
<td>5</td>
<td></td>
<td>10/16/91</td>
<td>18:15:35</td>
<td></td>
</tr>
<tr>
<td>BINLIST</td>
<td>@HTML</td>
<td>Z0</td>
<td>V</td>
<td>73</td>
<td>13</td>
<td></td>
<td>2/27/92</td>
<td>14:36:14</td>
<td></td>
</tr>
<tr>
<td>BINLIST</td>
<td>@INDEX</td>
<td>Z0</td>
<td>F</td>
<td>80</td>
<td>9</td>
<td></td>
<td>2/27/92</td>
<td>14:36:49</td>
<td></td>
</tr>
<tr>
<td>BINLIST</td>
<td>HELP</td>
<td>Z1</td>
<td>V</td>
<td>53</td>
<td>4</td>
<td></td>
<td>3/06/92</td>
<td>15:47:52</td>
<td></td>
</tr>
<tr>
<td>BINLIST</td>
<td>HTML</td>
<td>Z1</td>
<td>V</td>
<td>77</td>
<td>13</td>
<td></td>
<td>3/09/92</td>
<td>11:36:29</td>
<td></td>
</tr>
<tr>
<td>BINLIST</td>
<td>INDEX</td>
<td>Z1</td>
<td>F</td>
<td>80</td>
<td>10</td>
<td></td>
<td>2/27/92</td>
<td>14:54:47</td>
<td></td>
</tr>
<tr>
<td>BINLIST</td>
<td>SAVINDEX</td>
<td>Z1</td>
<td>F</td>
<td>80</td>
<td>8</td>
<td></td>
<td>2/27/92</td>
<td>14:36:37</td>
<td></td>
</tr>
<tr>
<td>BSDTYPES</td>
<td>H</td>
<td>Z1</td>
<td>V</td>
<td>70</td>
<td>120</td>
<td></td>
<td>10/15/91</td>
<td>18:01:30</td>
<td></td>
</tr>
<tr>
<td>CC_ALL</td>
<td>EXEC</td>
<td>Z1</td>
<td>V</td>
<td>81</td>
<td>13</td>
<td></td>
<td>10/14/91</td>
<td>15:29:29</td>
<td></td>
</tr>
<tr>
<td>CC2</td>
<td>EXEC</td>
<td>Z1</td>
<td>V</td>
<td>58</td>
<td>8</td>
<td></td>
<td>10/30/91</td>
<td>14:18:05</td>
<td></td>
</tr>
<tr>
<td>CMOD</td>
<td>EXEC</td>
<td>Z2</td>
<td>V</td>
<td>68</td>
<td>203</td>
<td></td>
<td>11/04/91</td>
<td>11:35:12</td>
<td></td>
</tr>
<tr>
<td>CMS</td>
<td>PRESERVE</td>
<td>Z1</td>
<td>V</td>
<td>65</td>
<td>30</td>
<td></td>
<td>11/06/91</td>
<td>17:06:02</td>
<td></td>
</tr>
<tr>
<td>CPOBJ</td>
<td>RMAP</td>
<td>Z1</td>
<td>F</td>
<td>80</td>
<td>3</td>
<td></td>
<td>10/16/91</td>
<td>18:46:59</td>
<td></td>
</tr>
<tr>
<td>DEFAULT</td>
<td>@HTML</td>
<td>Z0</td>
<td>V</td>
<td>81</td>
<td>13</td>
<td></td>
<td>12/06/91</td>
<td>21:15:48</td>
<td></td>
</tr>
<tr>
<td>DEFAULT</td>
<td>HTML</td>
<td>Z1</td>
<td>V</td>
<td>75</td>
<td>10</td>
<td></td>
<td>3/18/92</td>
<td>16:07:42</td>
<td></td>
</tr>
<tr>
<td>DEFAULTX</td>
<td>HTML</td>
<td>Z1</td>
<td>V</td>
<td>78</td>
<td>12</td>
<td></td>
<td>3/18/92</td>
<td>16:02:28</td>
<td></td>
</tr>
<tr>
<td>DLOAD</td>
<td>EXEC</td>
<td>Z1</td>
<td>V</td>
<td>64</td>
<td>4</td>
<td></td>
<td>11/01/91</td>
<td>15:15:36</td>
<td></td>
</tr>
<tr>
<td>EDCKV</td>
<td>C</td>
<td>Z1</td>
<td>F</td>
<td>80</td>
<td>5</td>
<td></td>
<td>10/16/91</td>
<td>18:08:43</td>
<td></td>
</tr>
<tr>
<td>EDCKV</td>
<td>TEXT</td>
<td>Z1</td>
<td>F</td>
<td>80</td>
<td>9</td>
<td></td>
<td>12/05/91</td>
<td>16:03:06</td>
<td></td>
</tr>
</tbody>
</table>
FEARCH ORIG Z1 V 76 125
FGET @EXEC Z0 V 60 16
FGET EXEC Z2 V 75 27
FGET ORIG Z2 V 78 132
FINDATE C Z1 V 78 245
FINDATE C2 Z1 V 78 245
FINDATE ORIG Z1 V 78 245
FINDATE SLACVM Z1 V 78 245
FINDATE TEXT Z1 F 80 113
FSEARCH @EXEC Z0 V 65 13
FSEARCH EXEC Z1 V 69 63
FSEARCH SAVE Z1 V 76 123
FSEARCH SAVEEXEC Z1 V 40 7
HEP @HTML Z0 V 74 9
HEP @INDEX Z0 V 43 12
HEP HELP Z1 V 71 27
HEP HTML Z1 V 73 9
HEP INDEX Z1 V 43 12
HTDAEMON C Z1 V 85 639
HTDAEMON C2 Z1 V 85 639
HTDAEMON MODULE Z1 V 65535 5
HTDAEMON TEXT Z1 F 80 183
HTTCP C Z1 V 79 282
HTTCP H Z1 V 78 94
HTTCP TEXT Z1 F 80 77
HTUTILS H Z1 V 79 85
LOAD MAP Z1 F 100 2
MAKE EXEC Z1 V 51 6
MYCC EXEC Z1 V 63 51
NICFOUN1 EXEC Z2 V 78 125
Q Q Z1 F 80 52
RUN EXEC Z1 V 74 21
RUNDAEMO EXEC Z1 V 33 5
SETUP EXEC Z1 V 62 4
SPIRES HTML Z1 V 74 9
SPIRES INDEX Z1 V 43 12
STORES HELP Z1 F 80 267
STORES HTML Z1 V 76 10
STORES INDEX Z1 F 80 10
STRING H Z2 F 80 75
TCP H Z1 V 91 180
TERRYH TYH239 Z1 F 132 87
d------> Batch return code = 0.

------> System cleanup $BCLEAN.

********************************************************************

* End execution of JCW122 for WINTERS at 03:15:11 on 22/07/98.
*
* Actual CPU time (mm:ss[.th]): Virtual = 0:00.51 , Total = 0:01.15
* I/O counts: SIO=000766 RDR=000000 PRT=000257 PCH=003755
*
* Space used on A-disk at job end: 48K
*
* I/O counts on units still available
* Spool units
* 0009: 180 000C: 0 000D: 89 000E: 0
* 00RE: 0
* Dad units


8/29/01
<table>
<thead>
<tr>
<th>Code</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0190(S)</td>
<td>34</td>
</tr>
<tr>
<td>0191(A)</td>
<td>100</td>
</tr>
<tr>
<td>019E(Y)</td>
<td>37</td>
</tr>
<tr>
<td>0555(Z)</td>
<td>194</td>
</tr>
<tr>
<td>059E(T)</td>
<td>12</td>
</tr>
<tr>
<td>059F(U)</td>
<td>6</td>
</tr>
<tr>
<td>05A0(V)</td>
<td>27</td>
</tr>
<tr>
<td>05A1(R)</td>
<td>18</td>
</tr>
</tbody>
</table>
-rw-r--r--  1 winters  sf
-r--r--r--  1 winters  sf
-r--r--r--  1 winters  sf
-r--r--r--  1 winters  sf
-r--r--r--  1 winters  sf
-r--r--r--  1 winters  sf
-r--r--r--  1 winters  sf
-rw-r--r--  1 winters  sf
-r--r--r--  1 winters  sf
-r--r--r--  1 winters  sf
-rw-r--r--  1 winters  sf

199 Dec 12 1991  spires.index
21627 Mar  5 1992  stores.help
313 Mar  9 1992  stores.html
810 Mar  4 1992  stores.index
6075 Oct 16 1991  string.h
5125 Oct 14 1991  tcp.h
11571 Nov 14 1991  terryh.tyh239
560 Jul 22 20:15  unix-commands
7236 Jul 21 20:33  winters.filelist
32452 Jul 21 20:18  winters.jcw122
0 Jul 22 03:39  winters.ls